

Hacking Decidim

```
~/projects/Decidim/decidim (0.19-stable)$ bundle exec rake development_app
create
create  README.md
create  Rakefile
create  .ruby-version
create  config.ru
create  .gitignore
create  Gemfile
run     git init from "."
S'ha inicialitzat un dipòsit buit del Git en /home/ivan/projects/Decidim/decidim/development_app/.git/
create  package.json
create  app
create  app/assets/config/manifest.js
create  app/assets/javascripts/application.js
create  app/assets/javascripts/cable.js
create  app/assets/stylesheets/application.css
create  app/channels/application_cable/channel.rb
create  app/channels/application_cable/connection.rb
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/jobs/application_job.rb
create  app/mailers/application_mailer.rb
create  app/models/application_record.rb
create  app/views/layouts/application.html.erb
create  app/views/layouts/mailer.html.erb
create  app/views/layouts/mailer.text.erb
```



PRO-TIP! You're not a hacker if you don't have your terminal in black

About me: Ivan Vergés

- Telecommunication Engineering (ETSETB/UPC)
- Working as a freelance developer and system maintainer since 2004
- PHP/Python/Ruby/Javascript
- 100% Linux & Free Software
- Lead developer in Platoniq since 2018
- I play the guitar



<https://twitter.com/ivanverges>



<https://meta.decidim.org/profiles/microstudi>



<https://github.com/microstudi>



ivan@platoniq.net

Decidim experience

- The unofficial guide to install Decidim:

<https://platoniq.github.io/decidim-install/>

- Adapting Decidim for organizations
- Custom modules (direct verifications)

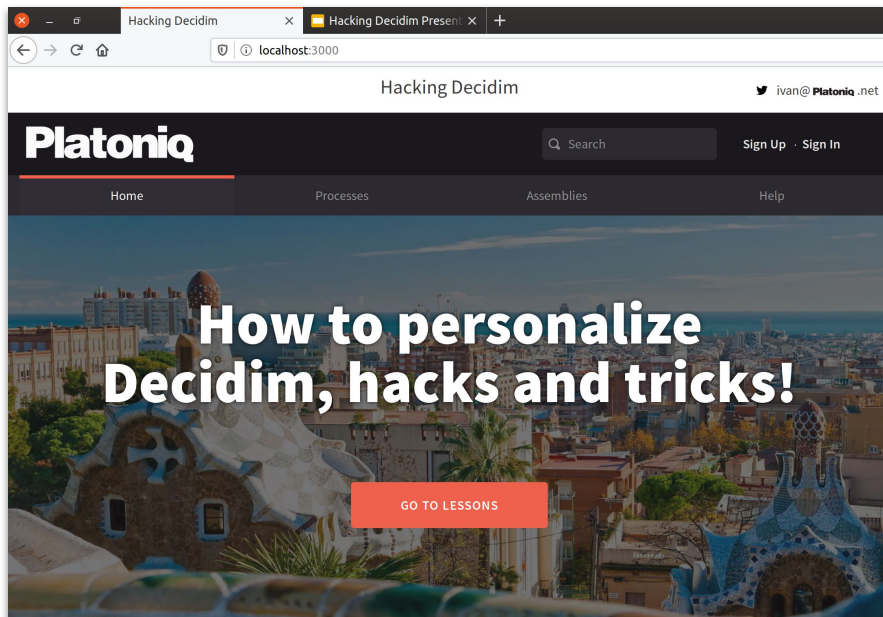
<https://github.com/Platoniq/>

- Among the contributors to the Decidim Source Code
- In the Team of maintainers of the source code

<https://github.com/decidim/decidim/graphs/contributors>

Requeriments: <https://github.com/platoniq/decidim-hacks>

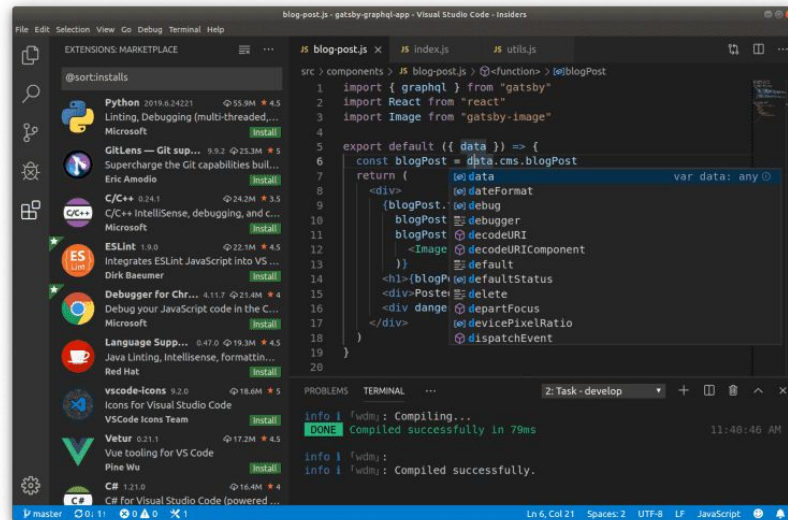
Docker up & Running with the test site ready
 docker-compose up



Learn how to hack and manipulate the default Decidim UI/UX and more

A proper text editor

<https://code.visualstudio.com/>



Install reminder

<https://code.visualstudio.com/>

<https://github.com/Platoniq/decidim-hacks/blob/master/docs/install-docker-on-windows-10-home.md>

<https://github.com/Platoniq/decidim-hacks/blob/master/docs/install-docker-on-macos.md>

```
git clone https://github.com/Platoniq/decidim-hacks.git
```

```
git pull
```

```
docker-compose up
```

Decidim-hacks is self-contained: <http://localhost:3000/processes>

HIGHLIGHTED PROCESSES

Basic Hacking examples (level 1)

This lesson covers some basics customizations.

[More info](#)

Hacking technique examples (level 2)

This lesson covers 5 basic techniques to change the default look & feel and behaviours of Decidim.

[More info](#)

```

32 <!-- Topbar -->
33 <div class="title-bar">
34 <div class="row column topbar">
35 <div class="logo-wrapper">
36 <img alt="decidim logo" data-bbox="400 760 430 780" class="img_header?"/>
37 </div>
38 </div>
39 </div>
40 <div class="row column topbar">
41 <div class="logo-wrapper">
42 <img alt="decidim logo" data-bbox="400 830 430 850" class="img_header?"/>
43 </div>
44 </div>
45 <div class="row column topbar">
46 <div class="logo-wrapper">
47 <img alt="decidim logo" data-bbox="400 880 430 900" class="img_header?"/>
48 </div>

```

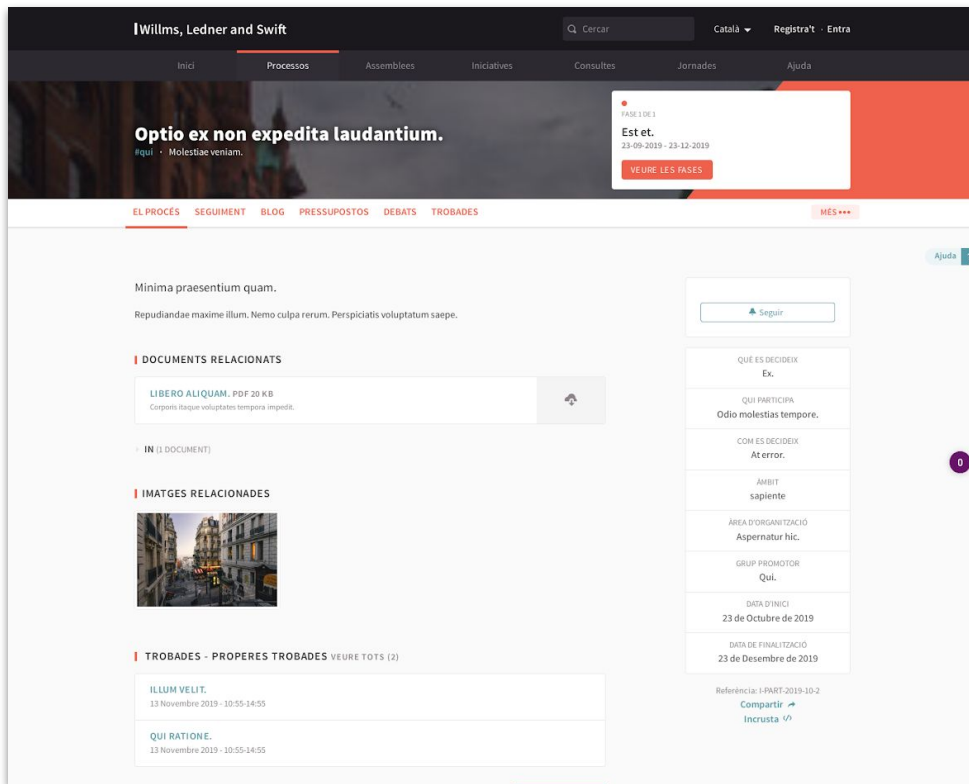
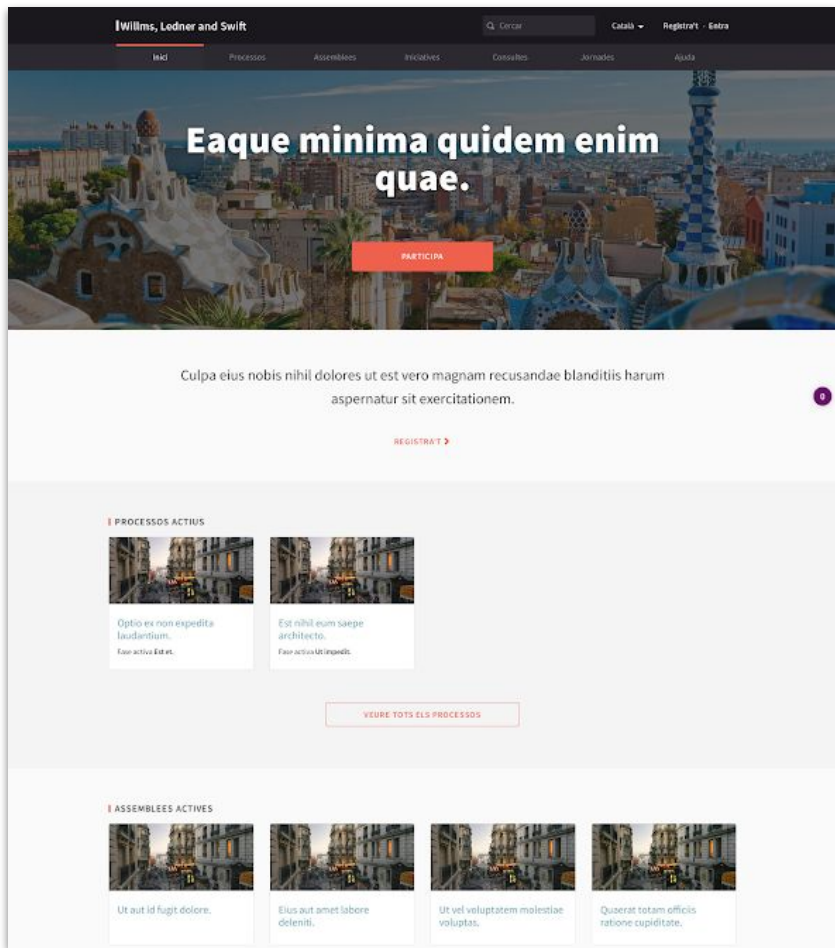
Find all the exercises here

Prepare environment

- A browser Tab with <http://localhost:3000/>
- A browser Tab with <https://github.com/Platoniq/decidim-hacks>
- Visual Studio Code opened, open the folder where you downloaded Decidim-hacks

Goals: what are we going to learn

- Get to know Decidim internally:
 - Directory structure, modular structure
- How to personalize Decidim in the frontend:
 - Change html, css and javascript defaults by our own implementations
- How to interact with the database directly:
 - Access the rails console
- How to change some backend behaviours:
 - Monkeypatch classes (overwrite methods)



How to turn this?

Voting is open 1-31.10.2019 - You can vote here ▶

Helsinki

Home Areas Information

Search

English

Sign In

OmaStadi

Participatory budgeting makes your ideas and voice heard

OmaStadi – Helsinki’s participatory budgeting



OmaStadi is Helsinki’s way of doing participatory budgeting. Helsinki uses €4.4 million annually for realising ideas developed by the residents of the city. You can bring up ideas and vote on omaStadi.hel.fi.

Vote!

Youth Ruuti Budget



Through the Ruuti Budget, young people can annually come up with ideas and negotiate things that concern them. The Ruuti Budget offers an opportunity to affect the development of youth services, leisure activities and the entire City.

Vote!

Vote and make a difference!

Between 1-31.10.2019, residents of Helsinki get to vote on OmaStadi plans made by the residents. The plans with the most votes will be realised.

Voting is open 1-31.10.2019 - You can vote here ▶

Helsinki

Home Areas Information

Search

English

Sign In

OmaStadi

Western Helsinki

#OmaStadi - Reijola, Munkkiniemi, Haaga, Pitäjämäki and Kaarola

PHASE 6 OF 7

Voting
2019-10-01 - 2019-10-31

View phases [Rajata](#)

INFO BUDGETS PLANS PROPOSALS 2018 EVENTS

The OmaStadi major district of Western Helsinki includes the neighbourhoods of Reijola, Munkkiniemi, Haaga, Pitäjämäki and Kaarola. Between 15 November and 9 December 2018, you can submit proposals on how the sum of €613,200 allocated for this area should be spent. What would make the district more functional, pleasant and fun?

Every year, the City of Helsinki will allocate €4.4 million of its budget to be decided upon by the city’s residents. The city is divided into seven major districts, and each district has been allocated a budget according to its population size. What would make the district more functional, pleasant and fun? You are not required to have accurate cost information in the proposal phase. The ideas are later developed into feasible plans by the residents and experts from the City services. The experts will create cost estimates for the plans. These plans are then voted on. The purpose of OmaStadi is to draw up proposals and make plans that are equal for all and benefit everyone. The district’s proposals can be voted on by everyone aged 12 or over. Submitting proposals, developing plans and voting are all done in the OmaStadi.hel.fi service.

The major district of Western Helsinki includes the following neighbourhoods:

- Reijola: Laakso, Vanha Ruskeasuo, Pikku Huopalahti, Mellahti
- Munkkiniemi: Niemenniemi, Munkkivuori, Talinranta, Vanha Munkkiniemi, Kuusisaari, Lehtisaari
- Haaga: Etelä-Haaga, Kivihaka, Pohjois-Haaga, Lassila, Pikku Huopalahti
- Pitäjämäki: Tali, Pajamäki, Pitäjämäki business zone, Reimaria, Marttila, Konala
- Kaarola: Kannelmäki, Maununeva, Malminkartano, Hakuniemi, Kuninkaantammi, Honkasuo.

Related images



[Follow](#)

START DATE
October 25, 2018

END DATE
October 31, 2019

Reference: HEL_PART_2018_10-10
[Share](#) [Embed](#)

Into this...

GOBIERNO DE LA CIUDAD DE MÉXICO Plaza Pública

Español - Regístrate | Entra

El espacio donde nos encontramos para decidir.

Plaza Pública es la plataforma de participación de la Ciudad de México donde la ciudadanía puede proponer, debatir, votar y decidir proyectos y disposiciones que afectan su vida en la ciudad.

Consultas

Vota en los temas puestos a consulta por parte del gobierno

[Ir a consultas](#)

Asambleas

Próximamente podrás involucrarte en un grupo de decisión temático.

[Ir a asambleas](#)

Iniciativas

Pronto podrás crear propuestas y firmas para mejorar la ciudad

[Ir a iniciativas](#)

¿Qué hacemos con la feria?

Las personas que habitamos y disfrutamos la Ciudad de México podemos opinar sobre **qué tipo de parque de diversiones queremos tener en el Bosque de Chapultepec.**

Las ideas y propuestas sobre qué tipo de atractivos debería tener, qué debería conservarse, qué hacer con la icónica montaña rusa o para qué sector de la población debería estar pensado, servirán de guía para las bases del concurso de licitación internacional que se abrirá próximamente para **decidir el futuro de ese preciado espacio de la Ciudad.**

El espacio de Chapultepec donde está la feria, **¿debe seguir siendo un parque de diversiones?**



Última actualización: 2019-10-21 10:35 hrs

Procesos terminados

Programa de Gobierno de la Ciudad de México 2019-2024

¿CÓMO ES LA LIMPIEZA EN EL CENTRO HISTÓRICO?

Diagnóstico Ciudadano ¿Cómo es la limpieza en el Centro Histórico?

Reconstrucción

Convocatoria para que personas damnificadas formen parte del Consejo Consultivo de la Comisión de Reconstrucción

COMISIÓN DE BÚSQUEDA DE LA CIUDAD DE MÉXICO

Elección de la persona titular de la Comisión de Búsqueda

GOBIERNO DE LA CIUDAD DE MÉXICO Plaza Pública

Español - Regístrate | Entra

FASE 3 DE 3

Incorporación de comentarios

2/6/2019 - 1/7/2019

[VER LAS FASES](#)

Programa de Gobierno de la Ciudad de México 2019-2024

#ProgramaDeGobiernoCDMX · Participa en la construcción de una Ciudad de innovación y derechos.

EL PROCESO PROGRAMA DE GOBIERNO 2019-2024

Conoce y comenta a favor, neutral o en contra de las acciones del Programa de Gobierno 2019-2024.

El Gobierno de la Ciudad de México pone a discusión pública el Programa de Gobierno 2019-2024 con el propósito de compartir y recoger la visión de la ciudadanía en la elaboración del documento que establece en los ejes, objetivos y acciones a realizar.

Este programa fue construido con las voces, plumas, opiniones de muchas personas y por supuesto, escuchando las opiniones de cientos de miles de ciudadanas y ciudadanos. Hoy nos toca gobernar para todos y para todas y por ello se pone a disposición de los habitantes de la Ciudad para su consulta y discusión.

La consulta estará abierta del 12 al 30 de abril.

Conocer la opinión y propuestas de la ciudadanía es clave en la búsqueda de consensos en torno a la construcción de la ciudad de innovación y derechos que queremos lograr.

[Seguir](#)

QUE SE DECIDE

Las acciones que integran el Programa de Gobierno.

QUIEN PARTICIPA

Habitantes de la Ciudad de México

CÓMO SE DECIDE

Comentando a favor, neutral o en contra de la propuesta.

ÁMBITO

Estatal

GRUPO PROMOTOR

Jefatura de Gobierno

FECHA DE INICIO

12 de abril de 2019

FECHA DE FINALIZACIÓN

1 de julio de 2019

Referencia: Plaza Pública - PART-2019-04-1

[Compartir](#)

[Incrustar](#)

DOCUMENTOS RELACIONADOS

RESULTADOS DE LA CONSULTA PDF 1000 KB

Resultados de la Consulta del Plan de Gobierno de la Ciudad de México 2019-2024

PROGRAMA DE GOBIERNO DE LA CIUDAD DE MÉXICO 2019-2024 PDF 20 MB

Programa de Gobierno de la Ciudad de México 2019-2024

PROGRAMA DE GOBIERNO 2019-2024 VER TODO (133)

VER TODO (133)

2.1.6. Mejorar la protección social para el desempleo

[Propuesta oficial](#)

Implementar mejoras en las políticas públicas de protección social dirigidas a las personas en...


10/04/2019	SEGUIR	0	0
------------	------------------------	---	---

5.1.8 Estrategia de seguridad para el transporte público

[Propuesta oficial](#)

Objetivo Mejorar las estrategias para disminuir la incidencia delictiva en el transporte públ...

10/04/2019	SEGUIR	0	1
------------	------------------------	---	---




[Kirjaudu sisään](#)
[Rekisteröidy](#)

[Etusivu](#)
[Prosessit](#)
[Tietoa](#)


Tervetuloa kehittämään Tuusulaa

[OSALLISTU](#)

Äktiiviset prosessit



Osallistuva budjetointi 2020
Aktiivinen vaihe **Idea-haku**



Osallistuva budjetointi Tuusulassa 2019
Aktiivinen vaihe **Tulosten julkaisu**

[NÄYTÄ KAIKKI PROSESSIT](#)

Kuinka osallistun?

TAPAHTUMAT

Selvitä missä ja milloin voit osallistua julkisiin tapahtumiin.

KESKUSTELUT


Välitele, keskustele ja jaa näkemyksesi rikastuttaksesi sinua koskettavia aiheita.

IDEAT

Tee ideoita, tue muiden tekemiä ideoita ja edistä muutoksia, joita haluat nähdä.

[LISÄTIETOJA](#)

Tervetuloa Tuusulan osallistumisolustalle!



OSALLISUUSALUSTA

[Kirjaudu sisään](#)
[Rekisteröidy](#)


[Etusivu](#)
[Osallistu](#)
[Tietoa](#)

TERVETULOA KEHITTÄMÄÄN YHDESSÄ KORSON SUURALUETTA!

[OSALLISTU](#)

MEIDÄN KORSO -HANKKEESSA KEHITETÄÄN KORSON SUURALUEEN TURVALLISUUTTA JA YHTEISÖLLISYYTTÄ. TULE MUKAAN IDEOIMAAN, KESKUSTELEMAAN, ÄÄNESTÄMÄÄN JA TOTEUTTAMAAN!

[REKISTERÖIDY >](#)



TULOSEN JULKISTUSJUHLA

Mitä Korson suuralueella tehdään kevään 2019 aikana turvallisuuden ja yhteisöllisyyden parantamiseksi? Korsolaiset ovat päässeet vaikuttamaan helmikuun alkupuolella äänestämällä, mikä ideat Meidän Korso -hankkeessa saavat toteutusrahaa.

Juhlimme voittajajoukkoa yhdessä keskiviikkona 6.3. klo 17 alkaen Lumon Villiänkassa. Juhlissa kuulet voittajajoukosta ja niiden toteuttamisesta sekä haluessasi voit ilmoittautua mukaan jonkin idean toteuttamiseen.

[OSALLISTU](#)

Goals: what are we NOT going to learn

- To code/program:
 - Ruby, Css, Javascript or Html
- Ruby & Rails
- To install Decidim

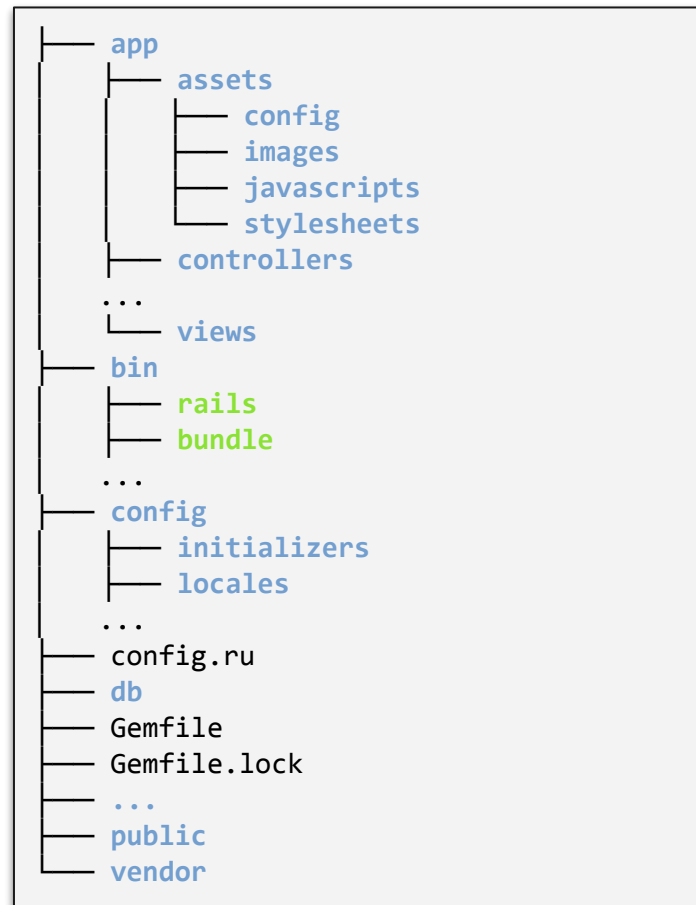
What is Decidim

- It's a **Gem** for the **Ruby and Rails** framework
- It is a **Generator** (like `rails new`).
- **It generates our app.**

```
~ $ decidim my_application
  create
  create README.md
  create Rakefile
  create .ruby-version
  create config.ru
  create .gitignore
  create Gemfile
  run git init from "."
```

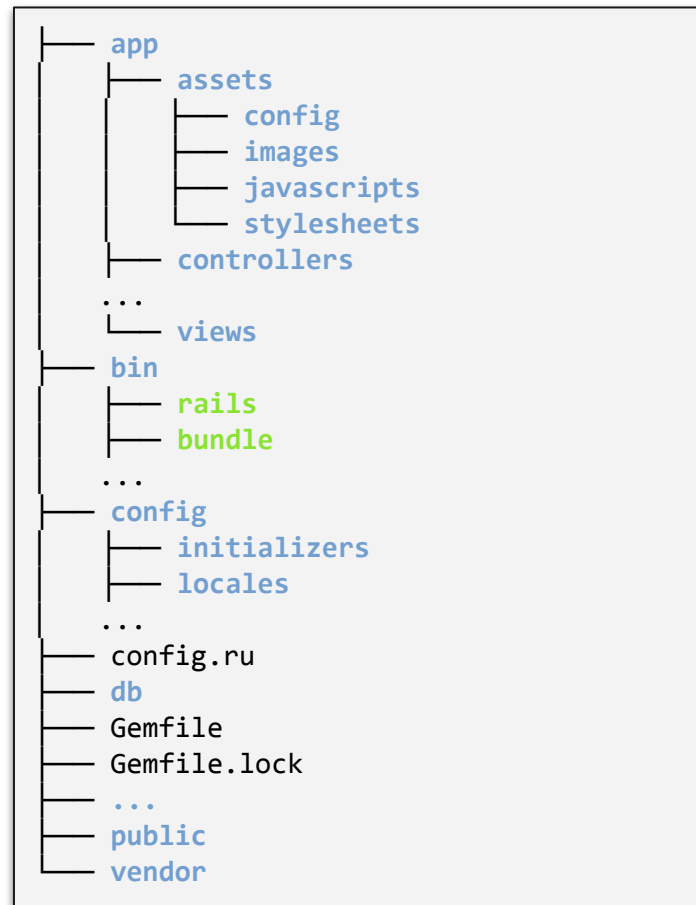
Decidim is Ruby & Rails

- **Gemfile** is the file used by **Bundler** to download all specified packages (gems) and its dependencies.
- **Gemfile.lock** contains all the gems installed with the exact version.
- **bundle**
 - installs Gems specified in the Gemfile
 - runs commands in the context of the Gemfile
- **rails**
 - Runs commands to manage Ruby & Rails:
 - bin/rails server
 - bin/rails console
 - bin/rails db:migrate

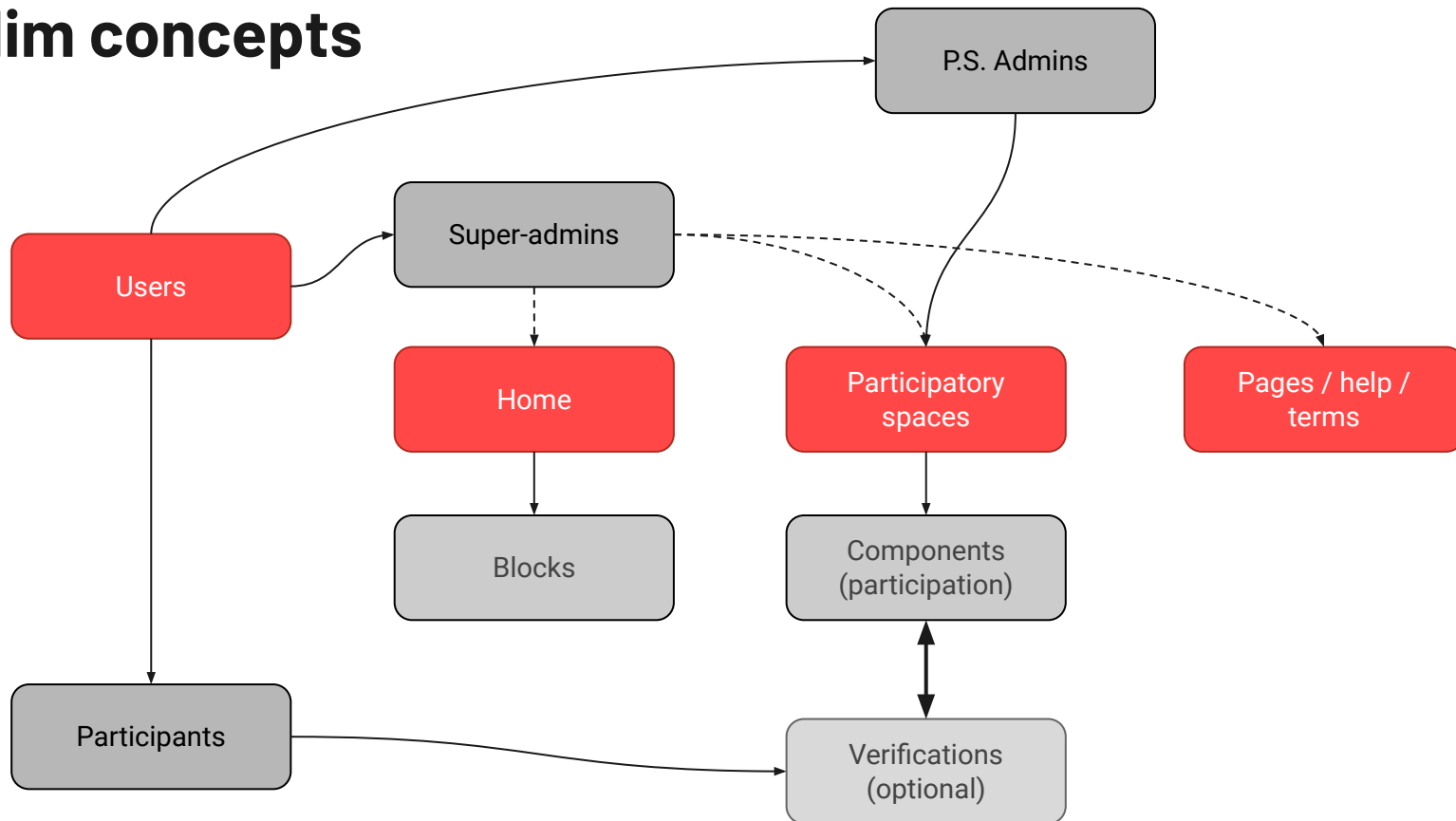


Decidim is Ruby & Rails

- **Rails** is (very) opinionated: it assumes there's one way to do most of the common things in the web.
- **"It's magic"**: it wants you to write the minimum amount of code. It's ok to not understand what's going on.
- It's for the Web and it's REST.
- Folder structure is immediately recognizable:
- In this workshop:
 - `app/assets/images`
 - `app/assets/javascripts`
 - `app/assets/stylesheets`
 - `views`
 - `config/initializers`
 - `config/locales`



Decidim concepts



Operational design for participation

P. Features

Participatory spaces

Components

C. Features

- Info
- **Phases**
- Categories
- Admins
- Components

- Info
- Categories
- **Members**
- Admins
- Components

Processes

Assemblies

Initiatives

...

Proposals

Meetings

Surveys

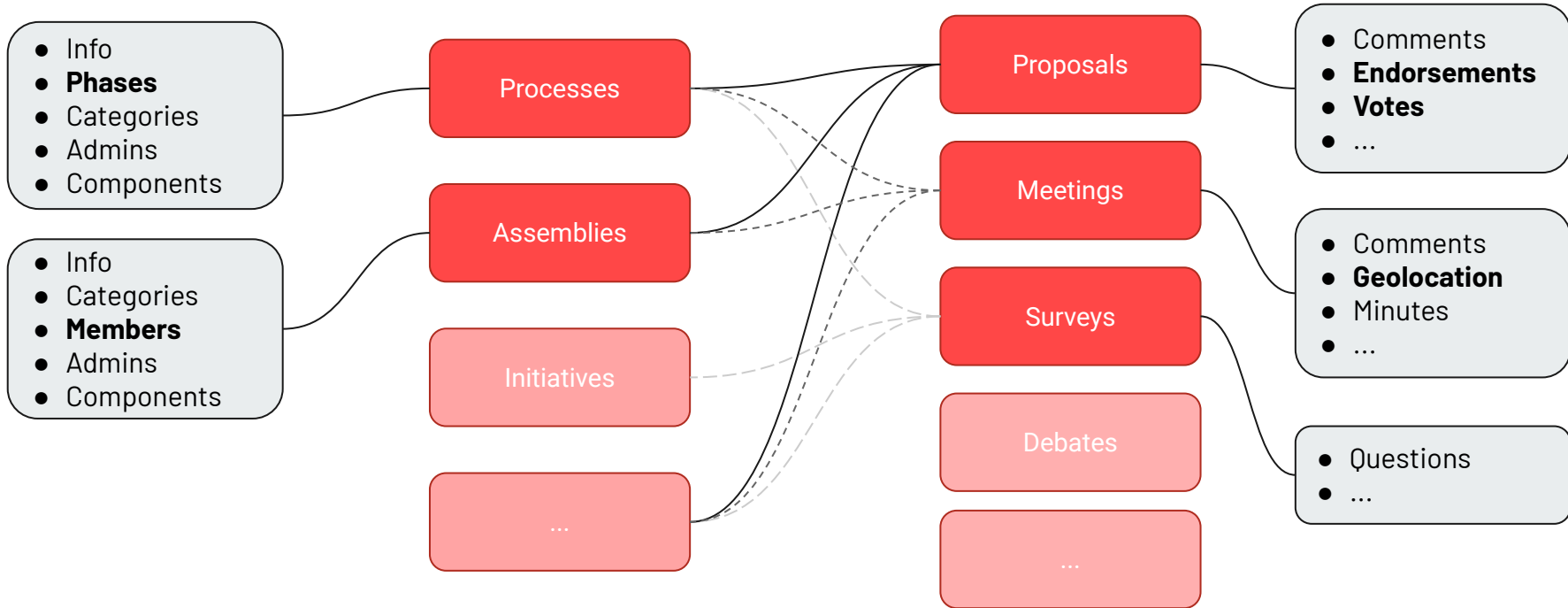
Debates

...

- Comments
- **Endorsements**
- **Votes**
- ...

- Comments
- **Geolocation**
- Minutes
- ...

- Questions
- ...



Co-design Platoniq's and ICLEI&FRIENDS's Sessions at Berlin's Consortium Assembly !

#openheritage · Co-design some of the activities in the next meeting of the consortium



PHASE 1 OF 2

Proposal definition

2019-04-14 - 2019-05-09

VIEW PHASES

THE PROCESS

DIGITAL COMMUNITY & PARTICIPATION

IMPROVING SOCIAL MEDIA ACTIVITIES

SOCIAL SURVEY FOR LABS

Info

Component

Phases

<https://labs.openheritage.eu/processes/berlin-meeting>

- DASHBOARD
- PROCESSES**
 - View public page
 - Info
 - Phases
 - Components**
 - Digital community & participation 5
 - Improving social media activities 9
 - Social Survey for Labs
- PROCESS GROUPS
- ASSEMBLIES
- CONFERENCES
- INITIATIVES
- CONSULTATIONS
- PAGES
- PARTICIPANTS

CO-DESIGN PLATONIQ'S AND ICLEI&FRIENDS'S SESSIONS AT BERLIN'S CONSORTIUM ASSEMBLY !

COMPONENTS ADD COMPONENT

COMPONENT NAME	COMPONENT TYPE	ACTIONS
Digital community & participation	Proposals	
Improving social media activities	Proposals	
Social Survey for Labs	Survey	

Process admin

Decidim modules

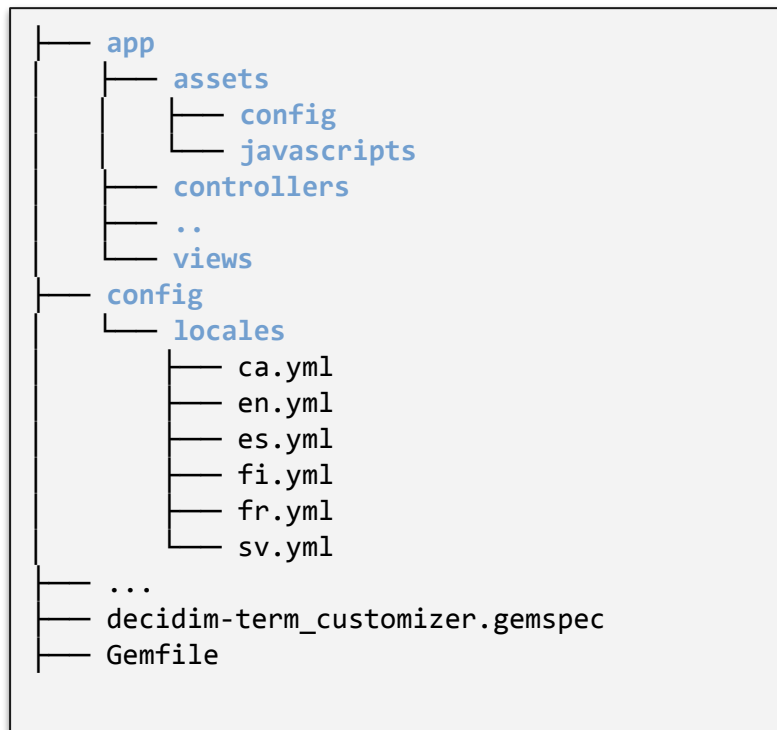
<https://github.com/decidim/decidim>

```
├── bin
├── config
├── d
├── decidim-accountability
├── decidim-admin
├── decidim-api
├── decidim_app-design
├── decidim-assemblies
├── decidim-blogs
├── decidim-budgets
├── decidim-comments
├── decidim-conferences
├── decidim-consultations
├── decidim-core
├── decidim-debates
├── decidim-dev
├── decidim-forms
├── decidim.gemspec
├── Gemfile
└── ...
```

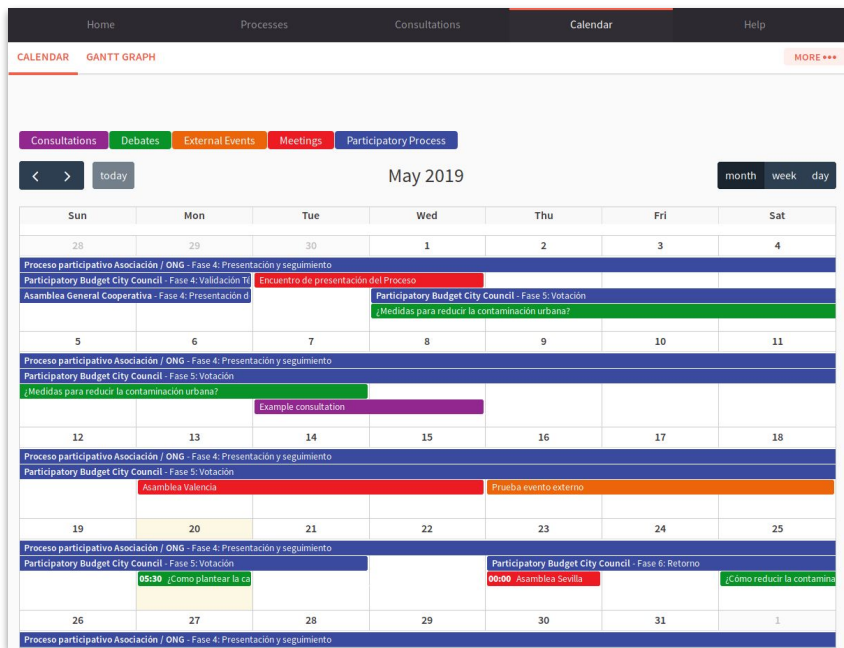
```
├── ...
├── decidim-accountability
│   ├── app
│   ├── bin
│   ├── config
│   ├── db
│   ├── decidim-accountability.gemspec
│   ├── lib
│   ├── Rakefile
│   ├── README.md
│   ├── spec
│   └── vendor
├── decidim-admin
│   ├── app
│   ├── assets
│   │   ├── config
│   │   ├── images
│   │   ├── javascripts
│   │   └── stylesheets
│   └── ...
└── ...
```

External modules

https://github.com/mainio/decidim-module-term_customizer



<https://github.com/alabs/decidim-module-calendar>



Technique 1: Overwrite Decidim HTML

- Views end in ***.erb**
- Rails looks for files in order when rendering a view (gems1 -> gem2 -> ... -> our app)
- We just need to replicate the exact path to overwrite in our **app/views** folder.
- Paths counts starting after the **app/views** part of any gem

Is not always that simple:

- Some content is generated with custom classes (Cells, Presenters, Content renderers...)
- In this case we need more advanced techniques (monkey patching)

Example 1: Change something in the main layout

Step 1: Find the view to overwrite. It may be defined on any module!

TIP Use the inspector tools in your browser to locate something unique (more or less)

Hacking Decidim

ivan@Platoniq.net

Platoniq

Search Sign Up Sign In

Home Processes Assemblies Help

How to personalize Decidim, hacks and tricks!

GO TO LESSONS

Learn how to hack and manipulate the default Decidim UI/UX and more

```

<html class="js" lang="en" ><script>
</script>
</head>
<body flex>
  <div class="platoniq-header">
    <script src="/assets/run_prettify.self.7285646e106490a20b5c3066f75eda638a6649ef84bbd5fb63122065bb668.js?bc">
    </script>
    <div class="off-canvas-wrapper">
      <div class="off-canvas-wrapper-inner" data-off-canvas-wrapper="">
        <div id="offCanvas" class="off-canvas position-right hide-for-large transition-push is-closed" data-position="right" data-off-canvas="" canvas="" aria-hidden="true">
          <div class="js-off-canvas-overlay is-overlay-flxed">
            <div class="off-canvas-content" data-off-canvas-content="">
              <div class="footer-separator">
                <!-- separates the footer from the rest of the page, creates a s -->
                </div>
                <div class="title-bar">
                  <div class="row column topbar">
                    <div class="logo-wrapper">
                      <a href="http://localhost:3000/">
                        <img alt="Hacking Decidim" src="/uploads/decidim/organiz

```


TIP Search in <https://github.com/decidim/decidim> for the located identifier

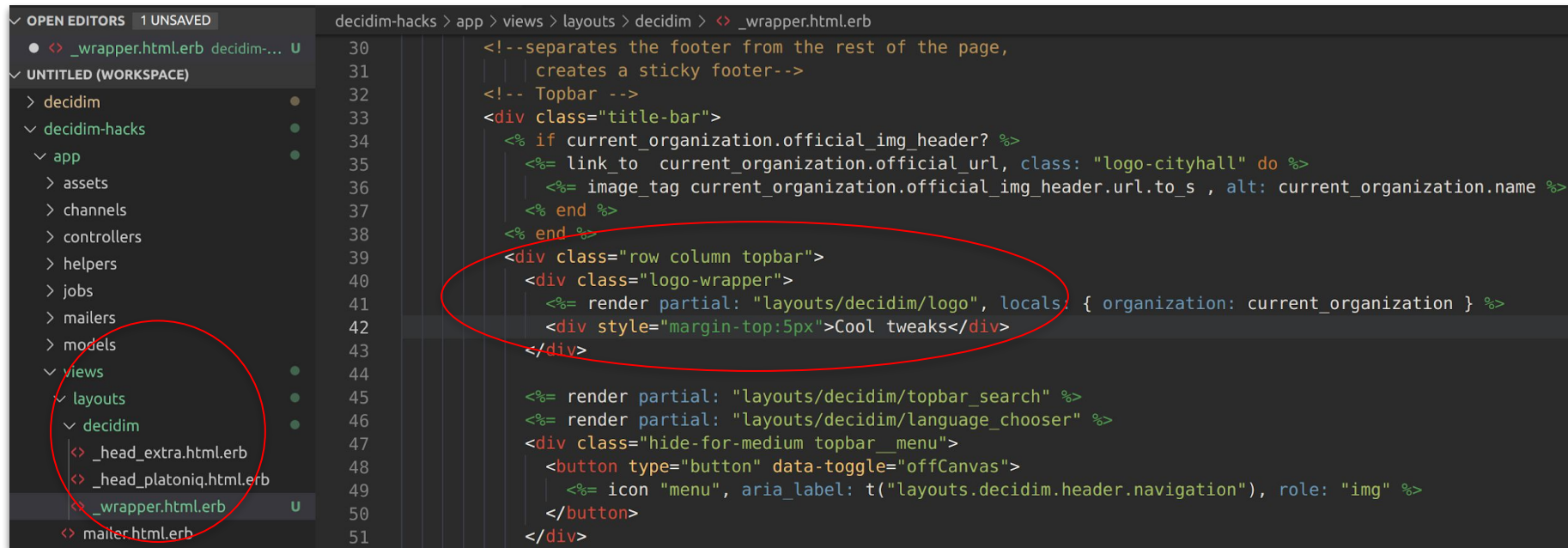
The screenshot shows a GitHub search interface for the repository `decidim/decidim`. The search term is `logo-wrapper`. The left sidebar shows search filters: Code (5), Commits (0), Issues (2), Packages (0), and Languages (HTML+ERB: 4, SCSS: 1). The main content area displays search results for the identifier `logo-wrapper`. The top result is from `decidim-core/app/views/layouts/decidim/_logo.html.erb`, showing code snippets with `.logo-wrapper` highlighted in yellow. A second result is partially visible below, showing `decidim-core/app/views/layouts/decidim/_wrapper.html.erb` with `logo-wrapper` also highlighted in yellow. The search results are filtered by `HTML+ERB`.

This is a zoomed-in view of the search results from the previous screenshot. It shows two search results for the identifier `logo-wrapper`. The top result is from `decidim-core/app/views/layouts/decidim/_logo.html.erb`, showing code snippets with `.logo-wrapper` highlighted in yellow. The second result is from `decidim-core/app/views/layouts/decidim/_wrapper.html.erb`, with the file path circled in red. This result shows code snippets with `logo-wrapper` highlighted in yellow. The search results are filtered by `HTML+ERB`.

PRO-TIP! Download the source code of Decidim, Open Visual Studio Code and search for that identifier locally!

Step 2:

1. Copy the file `decidim-core/app/views/layouts/decidim/_wrapper.html.erb` in your editor.
2. Change what you need.
3. Save it in your own place replicating the folder structure (`app/views/layouts/decidim/_wrapper.html.erb`)



```

decidim-hacks > app > views > layouts > decidim > <> _wrapper.html.erb
30 <!-- separates the footer from the rest of the page,
31      creates a sticky footer-->
32 <!-- Topbar -->
33 <div class="title-bar">
34   <%= if current_organization.official_img_header? %>
35     <%= link_to current_organization.official_url, class: "logo-cityhall" do %>
36       <%= image_tag current_organization.official_img_header.url.to_s , alt: current_organization.name %>
37     <%= end %>
38   <%= end %>
39   <div class="row column topbar">
40     <div class="logo-wrapper">
41       <%= render partial: "layouts/decidim/logo", locals: { organization: current_organization } %>
42       <div style="margin-top:5px">Cool tweaks</div>
43     </div>
44
45   <%= render partial: "layouts/decidim/topbar_search" %>
46   <%= render partial: "layouts/decidim/language_chooser" %>
47   <div class="hide-for-medium topbar__menu">
48     <button type="button" data-toggle="offCanvas">
49       <%= icon "menu", aria_label: t("layouts.decidim.header.navigation"), role: "img" %>
50     </button>
51   </div>
  
```

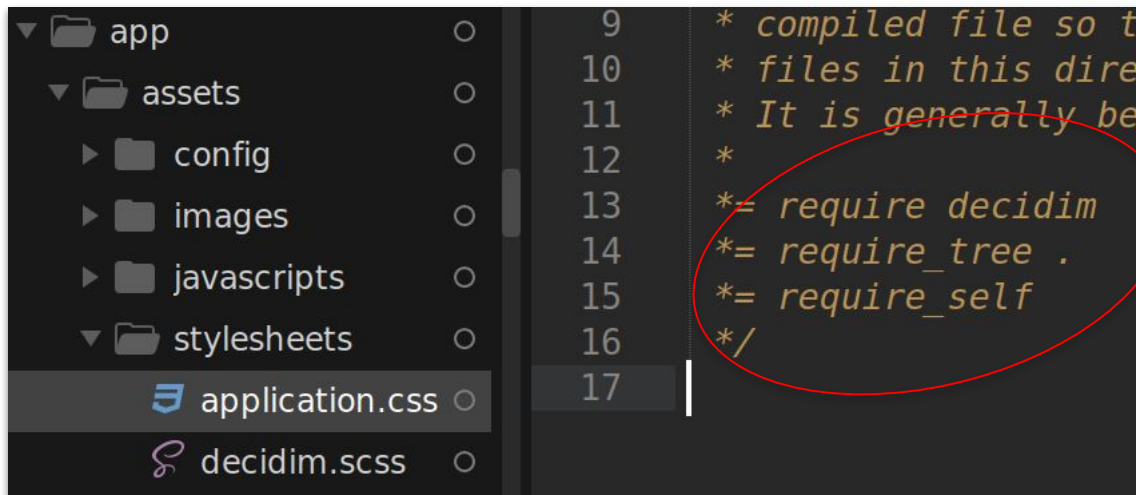
Step 3: Check in local!



PRO-TIP! Now is a good time to: `git commit -a -m "add cool tweaks text"`

Technique 2: Overwrite Decidim CSS

- Most CSS is SASS formatted (*.scss)
- Usually, we can overwrite partials the same way as HTML, the base route is now : **app/assets/stylesheets**
- How CSS is loaded is defined in the main **application.css** file:



```

9      * compiled file so th
10     * files in this direc
11     * It is generally be
12     *
13     *= require decidim
14     *= require_tree .
15     *= require_self
16     */
17

```

By default, the

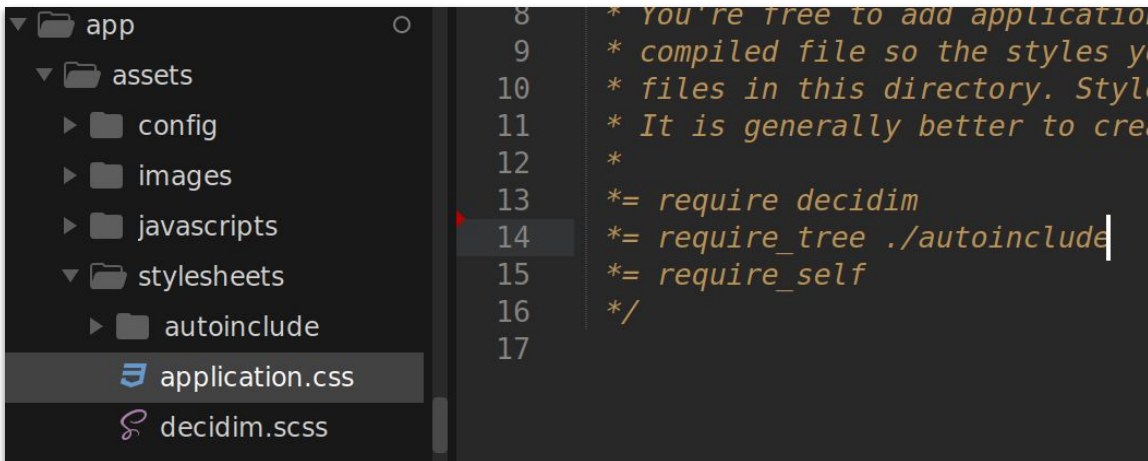
`require_tree .`

command loads everything
we have in that folder

Technique 2: Don't mess everything!

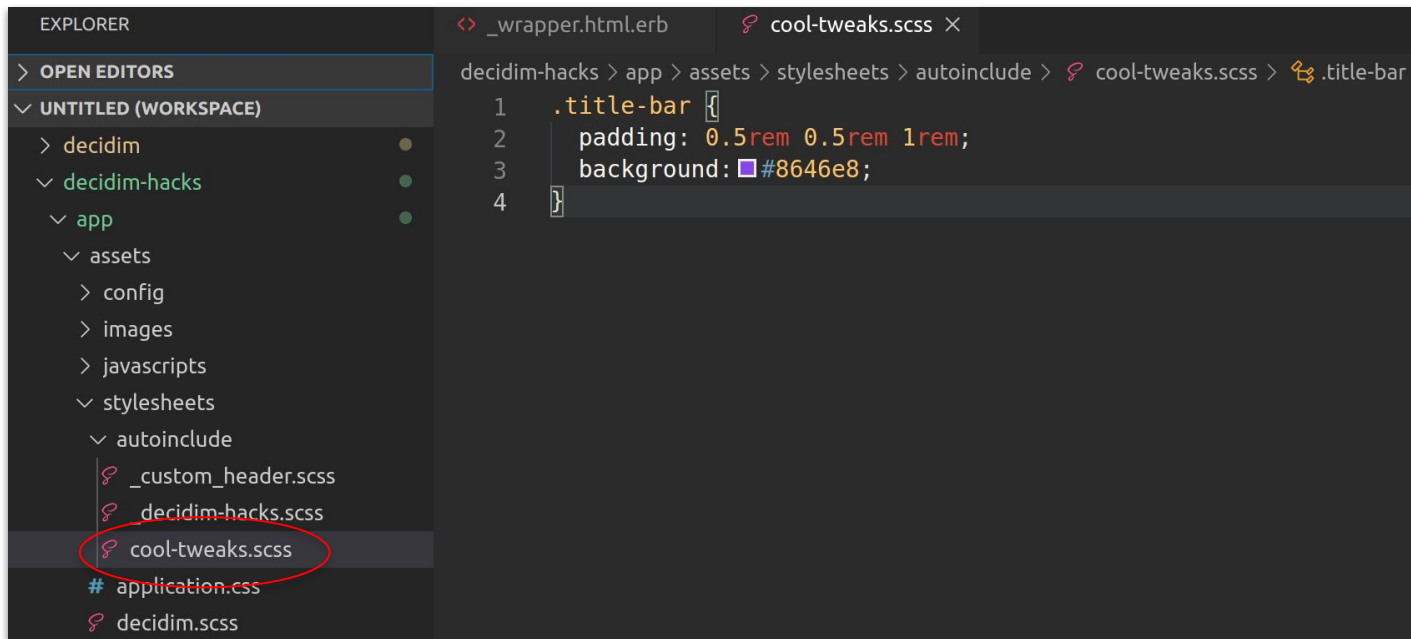
- Some colors are changed in settings!
- Most of the time we can just put our CSS on top of whatever exists
- We want to be able to mix our overwrites with our custom CSS

TIP: Modify the `require_tree` to a custom folder. Anything we will put there will be loaded automatically. We still have the chance to overwrite some specific view by using the duplicate path technique.



Step 2: Copy the changed CSS into a new file in our custom autoloading folder

```
padding: 0.5rem 0.5rem 1rem;
background: #8646e8;
```



The screenshot shows the VS Code interface. On the left, the Explorer sidebar shows the project structure with the file `cool-tweaks.scss` highlighted in red. On the right, the editor window shows the content of `cool-tweaks.scss` with the following CSS code:

```
decidim-hacks > app > assets > stylesheets > autoinclude > cool-tweaks.scss > .title-bar
1  .title-bar {
2    padding: 0.5rem 0.5rem 1rem;
3    background: #8646e8;
4  }
```

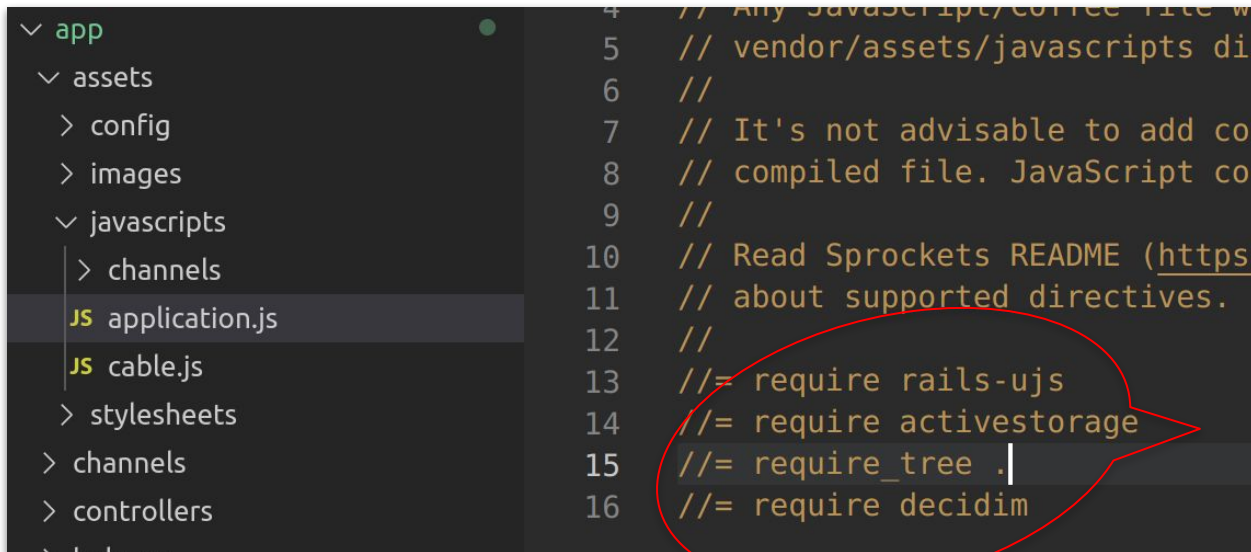
Step 3: Check in local!



PRO-TIP! You're not a hacker if you don't `git commit -a -m "add 90's background"`

Technique 3: Overwrite Decidim Javascript

- You can write in traditional Javascript or ES6 (*.js and *.js.es6)
- Javascript works similar as CSS, main route is now: **app/assets/javascripts**
- How Javascript is loaded is defined in the main **application.js** file:



```

4 // Any javascript/coffee file w
5 // vendor/assets/javascripts di
6 //
7 // It's not advisable to add co
8 // compiled file. JavaScript co
9 //
10 // Read Sprockets README (https
11 // about supported directives.
12 //
13 //= require rails-ujs
14 //= require activestorage
15 //= require_tree .
16 //= require decidim

```

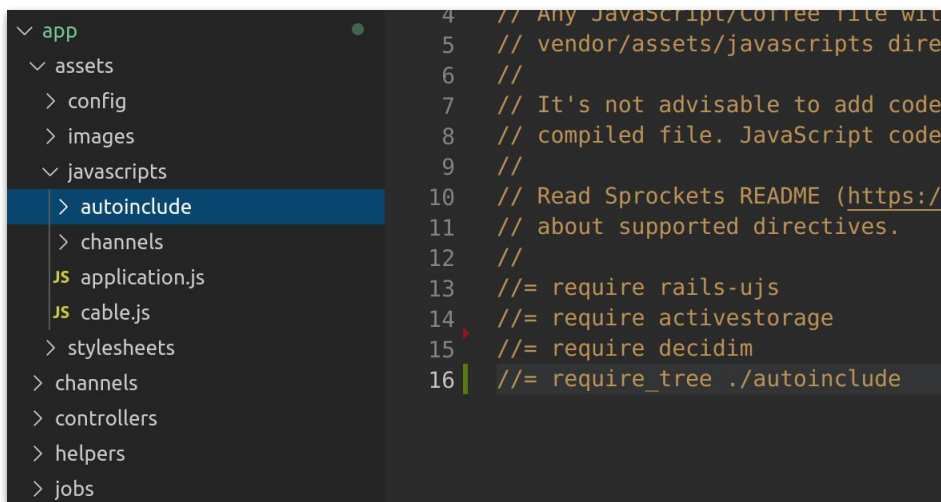
By default, the

`require_tree .`

command loads everything we have in that folder and is located BEFORE the Decidim javascript

Technique 3: Preparation

- If we add a new behaviour, we just add some additional javascript
- If we want to modify already existing scripts, we need to overwrite that file
- If we want to make sure that any custom script is applied AFTER Decidim, we need to change the loading order (add a custom autoloader folder for the sake of organization)



```

4 // Any Javascript/Coffee file wit
5 // vendor/assets/javascripts dire
6 //
7 // It's not advisable to add code
8 // compiled file. JavaScript code
9 //
10 // Read Sprockets README (https:/
11 // about supported directives.
12 //
13 //= require rails-ujs
14 //= require activestorage
15 //= require decidim
16 // require_tree ./autoinclude

```

Example 3: Add some Javascript

In this example we will apply also changes in CSS and HTML.

We will change the default behaviour of the main Hero's text image by:

1. Adding an external jQuery plugin
2. Overwriting the html view to show a custom text instead of the text from admin settings
3. Add a custom JS file



Super Customizable ¹Text Rotator with Style

PRO-TIP! Bad taste is a must for a hacker!

Step 1: Add the jQuery plugin to the system.

1. Download the jQuery plugin from:

<https://github.com/peachananr/simple-text-rotator>

- a. JS file: `vendor/assets/javascripts/jquery.simple-text-rotator.js`
- b. CSS file: `vendor/assets/stylesheets/simpletextrotator.css`

2. Edit `application.css` & `application.js` and tell them to require the downloaded files.

```

▼ stylesheets  ● 18
  ▼ autoinclude ● 19
    ? _custom_header.scss 21
    ? _decidim-hacks.scss 22
    ? cool-tweaks.scss     U 24
    # application.css      M 25
    ? decidim.scss        27
  * You're free to add application
  * compiled file so the styles y
  * files in this directory. Styl
  * It is generally better to cre
  *
  *= require decidim
  *= require simpletextrotator
  *= require_tree ./autoinclude
  *= require_self
  */

```

```

▼ javascripts  ● 13 // Read Sprockets README (https://githu
  ▼ autoinclude ● 14 // about supported directives.
    JS hero-fx.js U 15 //
    > channels 16 //
    JS application.js M 17 //
    JS cable.js 20 //
  //= require rails-ujs
  //= require activestorage
  //= require decidim
  //= require jquery.simple-text-rotator
  //= require_tree ./autoinclude

```

IMPORTANT! RELOAD THE SERVER NOW:

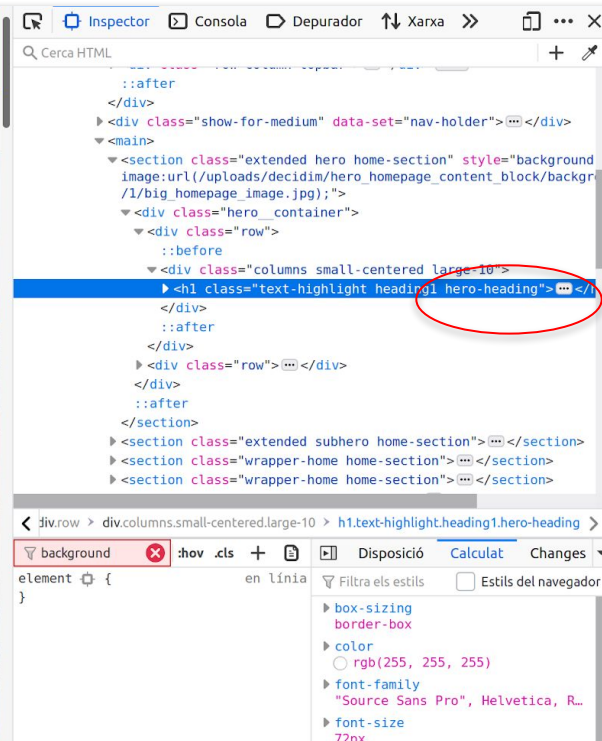
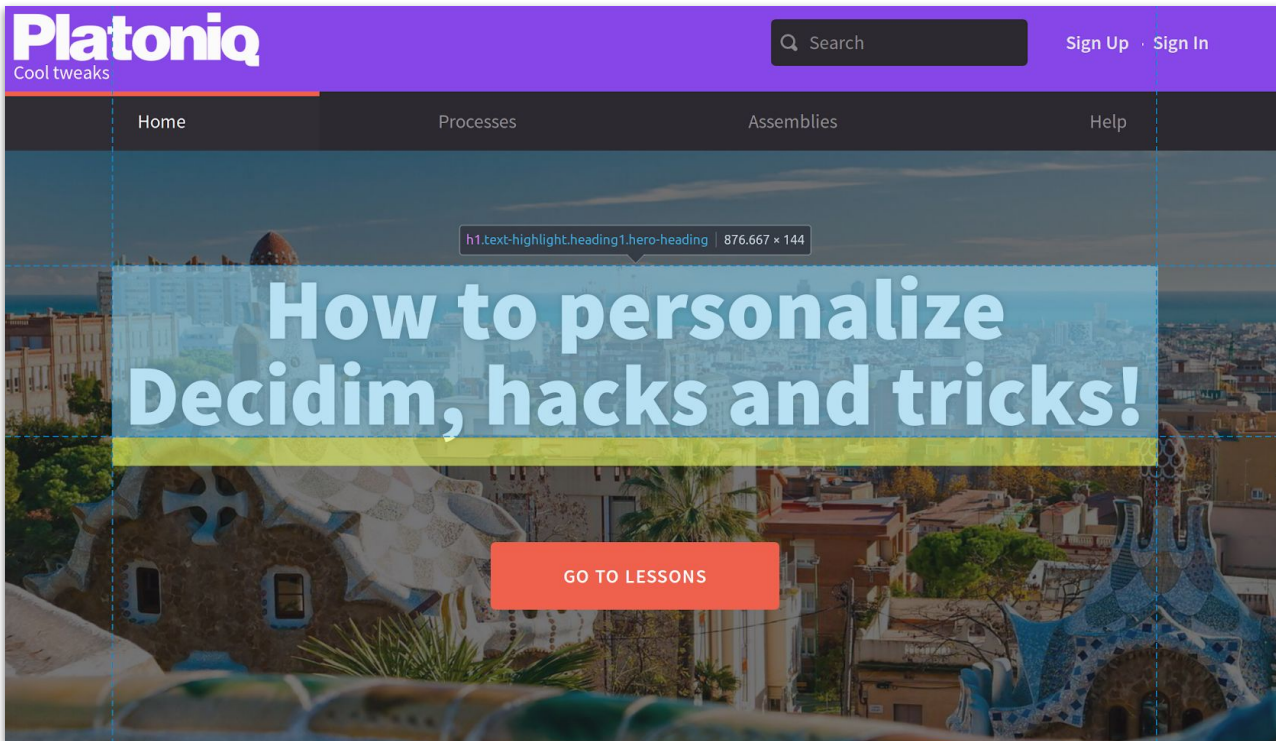
```

~ $ docker-compose exec app bin/rails restart

```

Step 2: Find the view to overwrite.

TIP Use the inspector tools in your browser to locate a useful identifier (ie: **hero-heading**)



hero-heading seems reasonably unique

NOTE: It is not a regular **html.erb** file!

The screenshot shows a GitHub search results page for the query 'hero-heading' in the 'decidim/decidim' repository. The browser address bar shows the search URL. The repository navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the left, a sidebar shows search filters: 'Code' (8 results), 'Commits' (0), 'Issues' (0), and 'Packages' (0). Below this, a 'Languages' section lists 'HTML+ERB' (7) and 'SCSS' (1). At the bottom of the sidebar are links for 'Advanced search' and 'Cheat sheet'. The main content area displays '8 code results in decidim/decidim' with a link to view all results. The first result is 'decidim-core/app/cells/decidim/content_blocks/hero/show.erb', which is circled in red. Its code snippet shows an ERB template with a highlighted 'hero-heading' class. The second result is 'decidim_app-design/app/views/public/partials/_conferences_hero.html.erb', also with a highlighted 'hero-heading' class. Both results indicate they are HTML+ERB files and show their last indexed dates.

Step 3: Overwrite the original view by creating a new one.

1. Replicate folder structure and copy the original file:

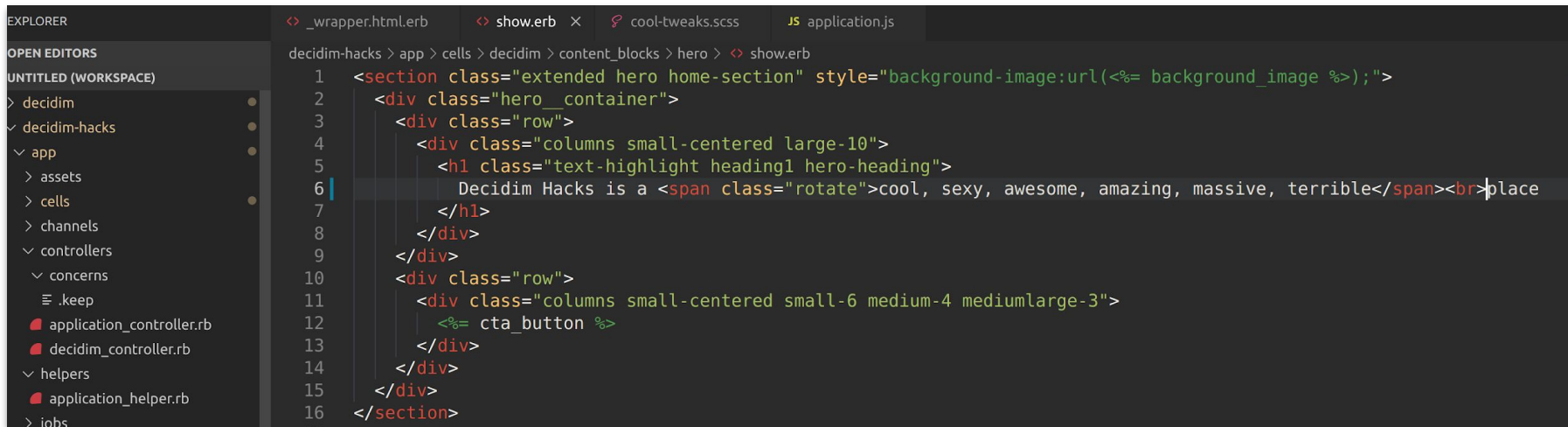
```
decidim-core/app/cells/decidim/content_blocks/hero/show.erb
```



2. In our folder: `app/cells/decidim/content_blocks/hero/show.erb`

Step 4: Edit the file. app/cells/decidim/content_blocks/hero/show.erb

We will remove any string coming from the database (sorry admins, no more Hero editing for you) and put a fixed phrase:



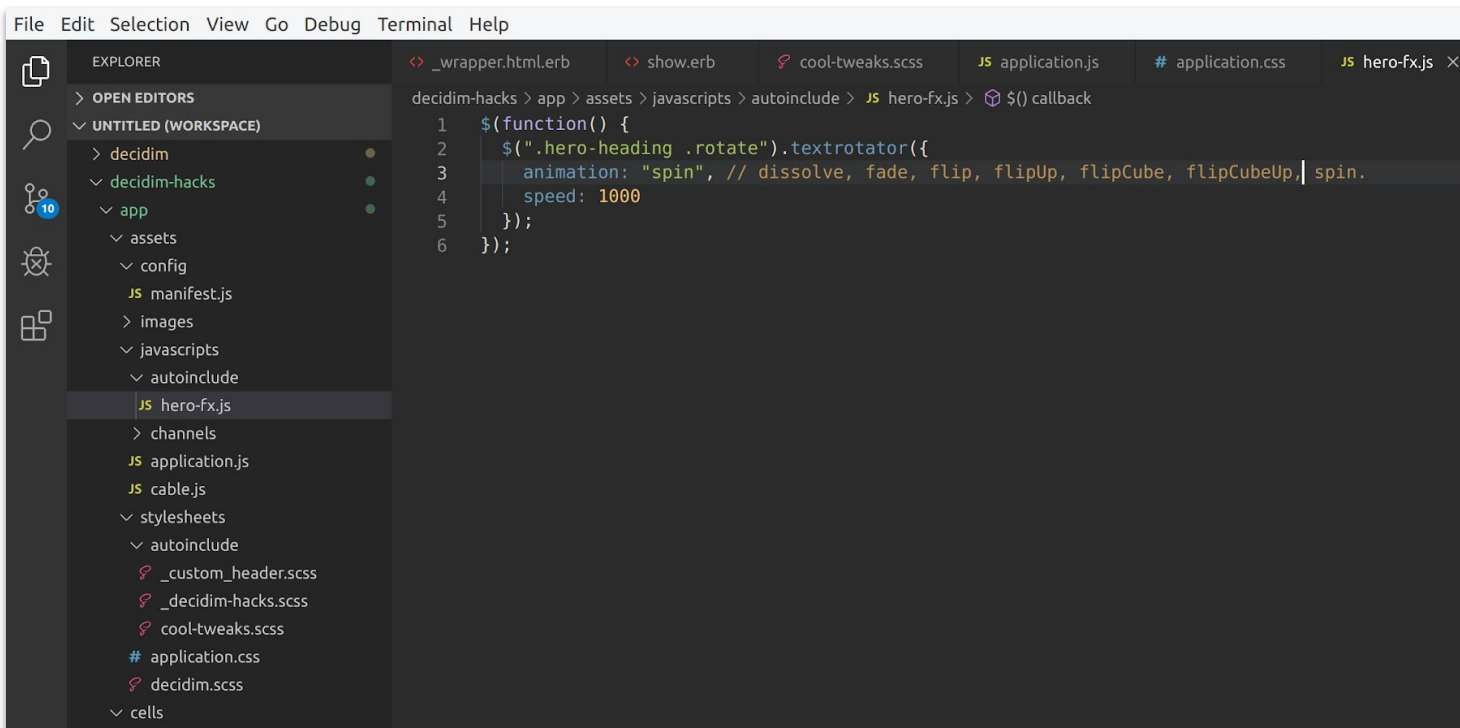
```

1 <section class="extended hero home-section" style="background-image:url(<%= background_image %>);">
2   <div class="hero_container">
3     <div class="row">
4       <div class="columns small-centered large-10">
5         <h1 class="text-highlight heading1 hero-heading">
6           Decidim Hacks is a <span class="rotate">cool, sexy, awesome, amazing, massive, terrible</span><br>place
7         </h1>
8       </div>
9     </div>
10    <div class="row">
11      <div class="columns small-centered small-6 medium-4 mediumlarge-3">
12        <%= cta_button %>
13      </div>
14    </div>
15  </div>
16 </section>

```

Decidim Hacks is a cool, sexy,
awesome, amazing, massive, terrible
place

Step 5: Create your own javascript that initializes everything, for instance:
 app/assets/javascripts/autoinclude/hero-fx.js

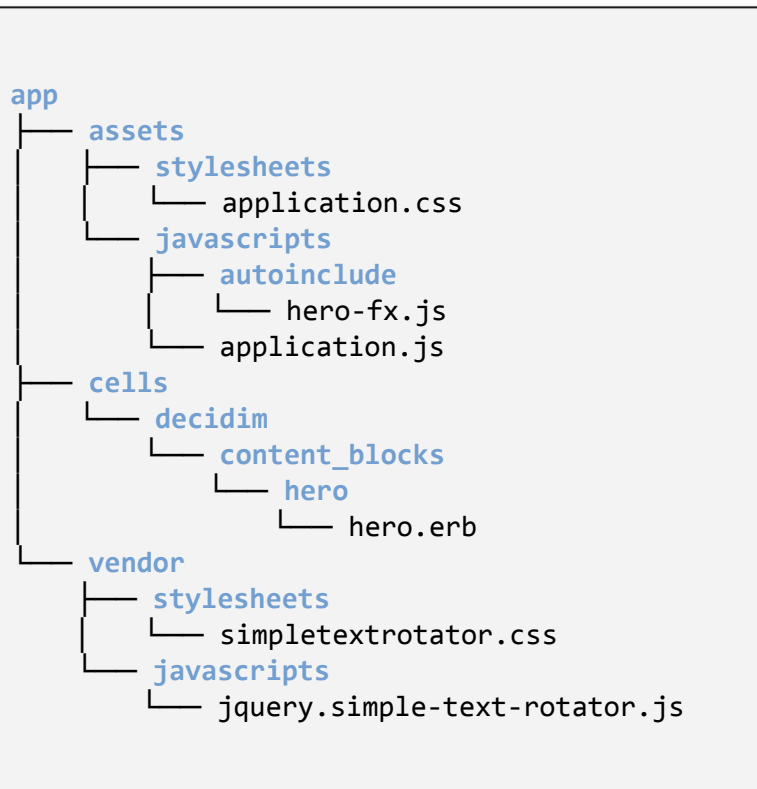


The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure with the file `hero-fx.js` selected under `app/assets/javascripts/autoinclude/`. The Editor window shows the content of `hero-fx.js`:

```
decidim-hacks > app > assets > javascripts > autoinclude > JS hero-fx.js > ($) callback
1  $(function() {
2    $(".hero-heading .rotate").textrotator({
3      animation: "spin", // dissolve, fade, flip, flipUp, flipCube, flipCubeUp, spin.
4      speed: 1000
5    });
6  });
```

Recapitulation:

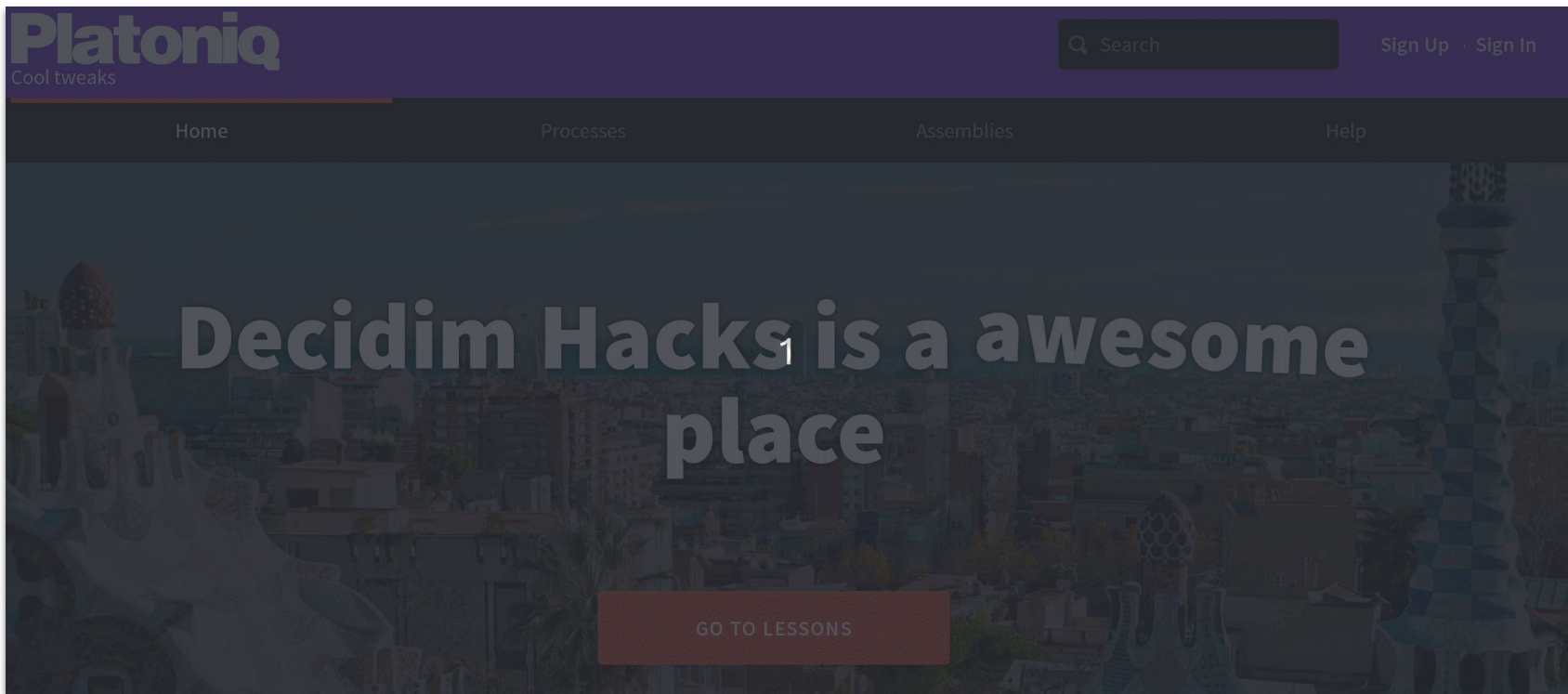
Files changed:



Steps:

1. Added external (vendor) javascript/css to the project
2. Configured Rails to load those files
3. Restarted Rails
4. Overwritten the Hero view.
5. Applied some custom javascript (activate the plugin)

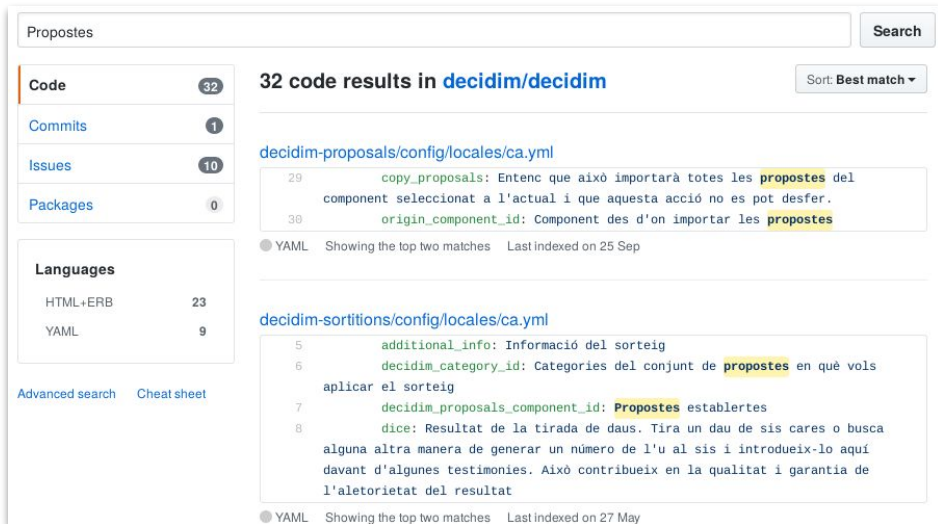
Step 6: Enjoy!



PRO-TIP! Hackers like to `git commit -a -m "add the coolest fx ever"`

Technique 4: Overwrite locales

- Locales are the multi-language systems texts
- Written in YAML format.
- Inside **app/config/locales/*.yml** (every module/gem has its owns).
- To overwrite, just create the same key with different content (for each language active!)



Propostes Search

Code 32

Commits 1

Issues 10

Packages 0

Languages

HTML+ERB	23
YAML	9

[Advanced search](#) [Cheat sheet](#)

32 code results in decidim/decidim Sort: Best match ▾

[decidim-proposals/config/locales/ca.yml](#)

```

29   copy_proposals: Entenc que això importarà totes les propostes del
30   component seleccionat a l'actual i que aquesta acció no es pot desfer.
31   origin_component_id: Component des d'on importar les propostes

```

● YAML Showing the top two matches Last indexed on 25 Sep

[decidim-sortitions/config/locales/ca.yml](#)

```

5   additional_info: Informació del sorteig
6   decidim_category_id: Categories del conjunt de propostes en què vols
7   aplicar el sorteig
8   decidim_proposals_component_id: Propostes establertes
9   dice: Resultat de la tirada de daus. Tira un dau de sis cares o busca
10  alguna altra manera de generar un número de l'u al sis i introdueix-lo aquí
11  davant d'algunes testimonies. Això contribueix en la qualitat i garantia de
12  l'aleatorietat del resultat

```

● YAML Showing the top two matches Last indexed on 27 May

Example 4: Add custom locales

In this example we will create a new FX like the previous example to use multi-language texts.

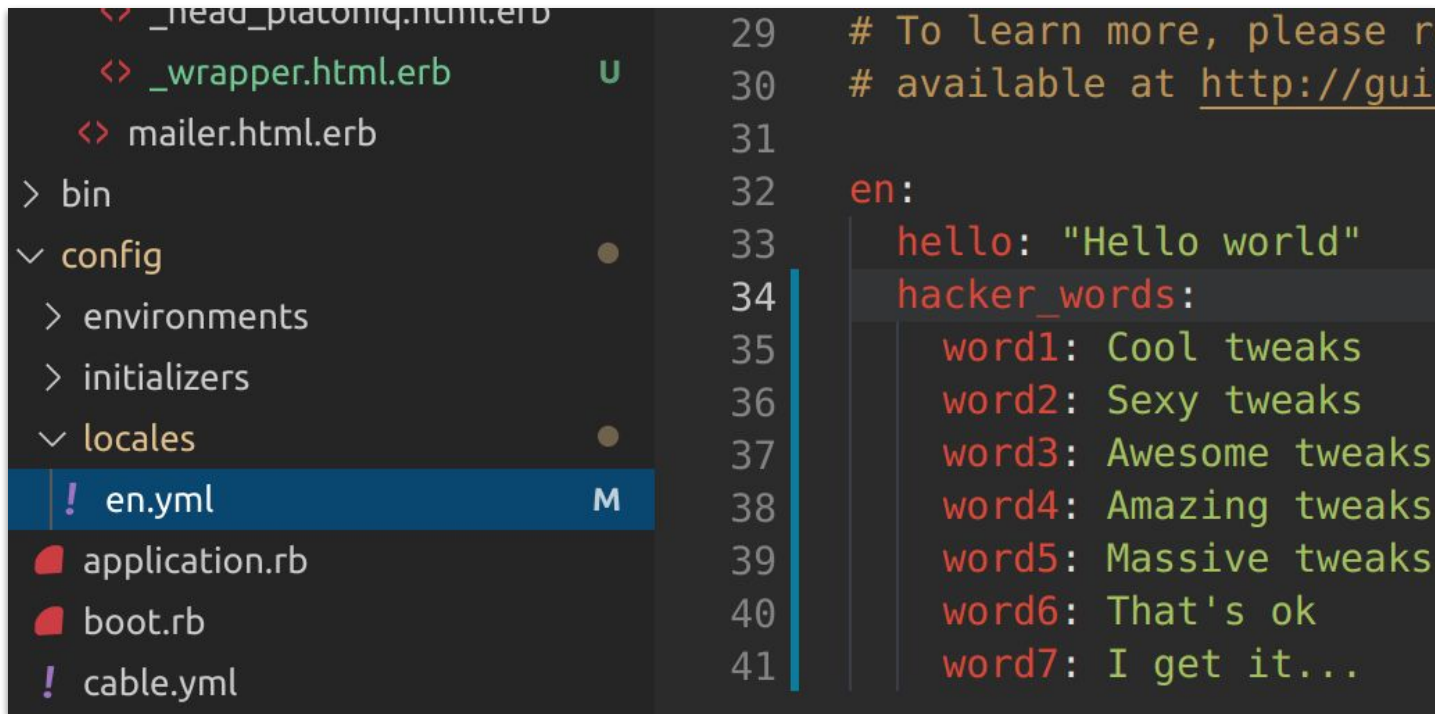
We will:

1. Create the list of words that rotate as independent text entries in the app locales folder.
2. Create a "helper" ruby method that will list the words for us in the view.
3. Add a new javascript file to take care of this case.

NOTE: Helpers cannot be defined this way in the previous example (it is not a regular rails **html.erb** file).

Step 1: Edit the locales file: `config/locales/en.yml`

Add some content in the YAML format:



```

    <> _head_platoniq.html.erb      29
    <> _wrapper.html.erb           30
    <> mailer.html.erb             31
  > bin                           32
  <v> config                       33
    > environments                 34
    > initializers                 35
    <v> locales                     36
      ! en.yml                     37
      application.rb               38
      boot.rb                      39
      ! cable.yml                  40
                                  41
# To learn more, please refer to the Rails guides, which are
# available at http://guides.rubyonrails.org

en:
  hello: "Hello world"
  hacker_words:
    word1: Cool tweaks
    word2: Sexy tweaks
    word3: Awesome tweaks
    word4: Amazing tweaks
    word5: Massive tweaks
    word6: That's ok
    word7: I get it...
  
```

Step 2: Edit the layout file: `app/views/layouts/decidim/_wrapper.html.erb`

Change the previous “Cool tweaks” for the method we are going to use:

```

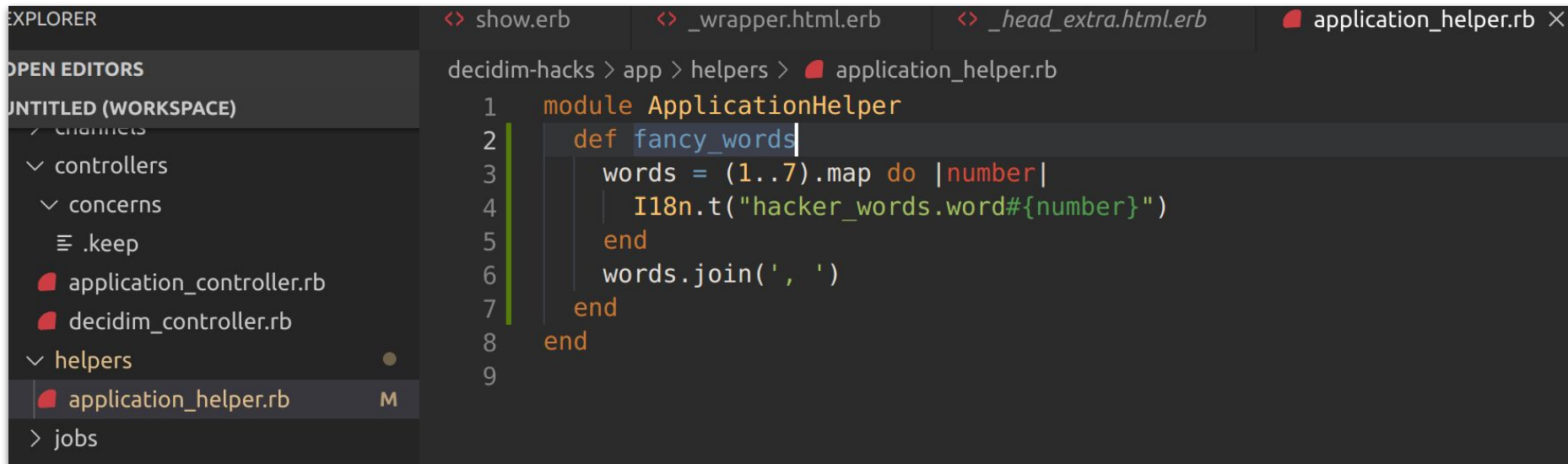
  38
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  <% end %>
  <div class="row column topbar">
    <div class="logo-wrapper">
      <%= render partial: "layouts/decidim/logo", locals: { organization: cu
      <div class="rotate" style="margin-top:5px"><%= fancy_words %></div>
    </div>

    <%= render partial: "layouts/decidim/topbar_search" %>
    <%= render partial: "layouts/decidim/language_chooser" %>
    <div class="hide-for-medium topbar__menu">
      <button type="button" data-toggle="offCanvas">
        <%= icon "menu", aria_label: t("layouts.decidim.header.navigation"),
      </button>
    </div>

```

Step 3: Edit the helper's file: `app/helpers/application_helpers.rb`

Create the method `fancy_words`:



The screenshot shows a code editor with the following content:

```

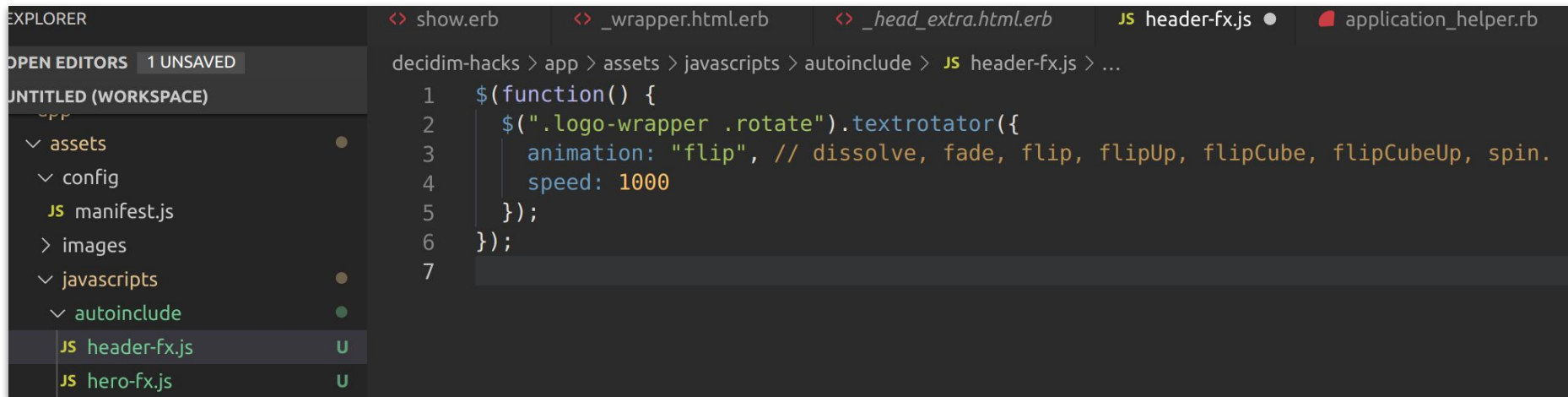
EXPLORER
  OPEN EDITORS
  UNTITLED (WORKSPACE)
  channels
  controllers
  concerns
  .keep
  application_controller.rb
  decidim_controller.rb
  helpers
    application_helper.rb M
  jobs

  show.erb
  _wrapper.html.erb
  _head_extra.html.erb
  application_helper.rb X

decidim-hacks > app > helpers > application_helper.rb
1  module ApplicationHelper
2      def fancy_words
3          words = (1..7).map do |number|
4              I18n.t("hacker_words.word#{number}")
5          end
6          words.join(', ')
7      end
8  end
9

```


Step 4: Create the javascript associated file to initialize the plugin:
app/assets/javascripts/autoinclude/header-fx.js:



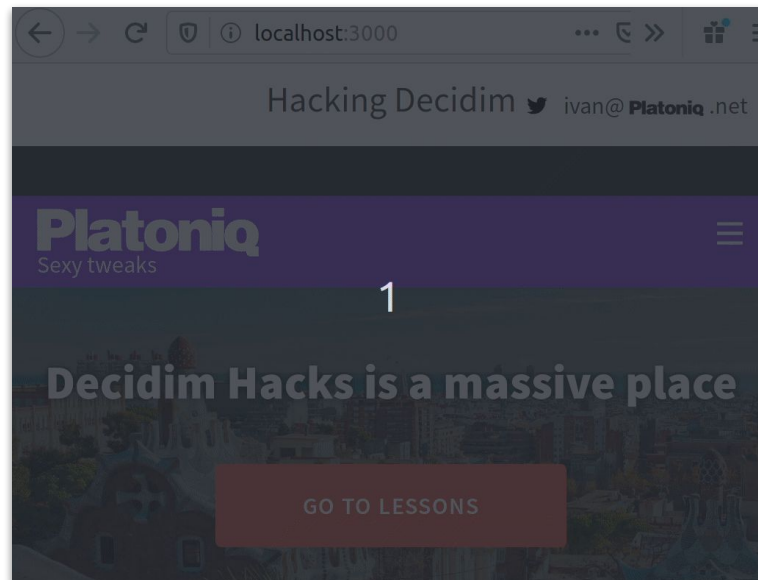
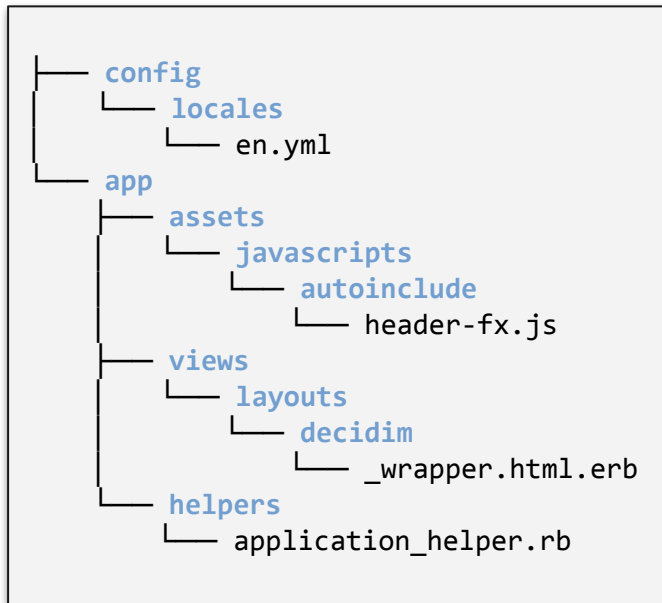
The screenshot shows a code editor interface. On the left is the Explorer sidebar with the following structure:

- EXPLORED
- OPEN EDITORS 1 UNSAVED
- UNTITLED (WORKSPACE)
- assets
- config
- JS manifest.js
- images
- javascripts
 - autoinclude
 - JS header-fx.js U
 - JS hero-fx.js U

The main editor area shows the content of `header-fx.js` with the following code:

```
decidim-hacks > app > assets > javascripts > autoinclude > JS header-fx.js > ...
1  $(function() {
2    $(".logo-wrapper .rotate").textrotator({
3      animation: "flip", // dissolve, fade, flip, flipUp, flipCube, flipCubeUp, spin.
4      speed: 1000
5    });
6  });
7
```

Step 5: Check and recapitulate



PRO-TIP! Hackers are humans too... `git commit -a -m "🙏 please no more effects..."`

Technique 5: Monkey Patching

Metaprogramming

- Ruby is an extremely manipulable programming language
- Almost everything can be redefined.
- We will use this capability to redefine methods defined in the Decidim core without having to touch it

Hooks

- Ruby (as a language) has hooks when certain events happen
- `class_eval` for creating instance methods
- `instance_eval` for creating class methods



What is monkey patching?

From the Decidim gem, loaded first

```
1 Decidim
2   def say_hello
3     "hello"
4   end
5 end
```

In our app, loaded after Decidim

```
1 Decidim
2   def say_hello
3     "bye"
4   end
5 end
```

Where to monkey patch in Decidim

PROBLEM: Most of the classes are already instantiated when Rails load our app.

- We will use Ruby hooks to append our code in a custom Rails initializer, usually `class_eval`
- A Rails initializer is just a Ruby file inside `config/initializers/something.rb`. It's evaluated when the application starts (once only)
- Changing a initializer requires Rails reload

Example 5: Overwrite non-configurable settings

In this example we will monkey-patch the Proposals method that check the titles of the proposals

We will:

1. Identify the Proposal validation method
2. Create a new initializer and overwrite the method to allow more flexible titles in proposals

< Back

CREATE YOUR PROPOSAL

You are creating a proposal.

Create your proposal

Compare

Complete

Publish your proposal

Title *

...

i am short

must start with a capital letter, is too short (under 15 characters)

Body *

...

Hello, I want to hack!

Create proposal as

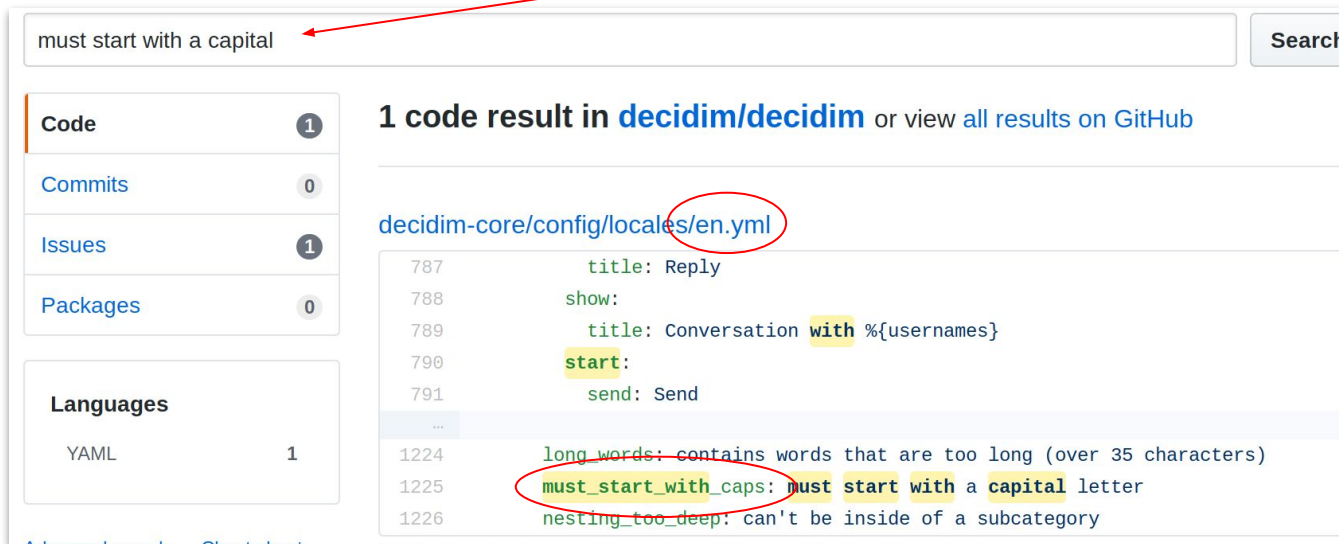
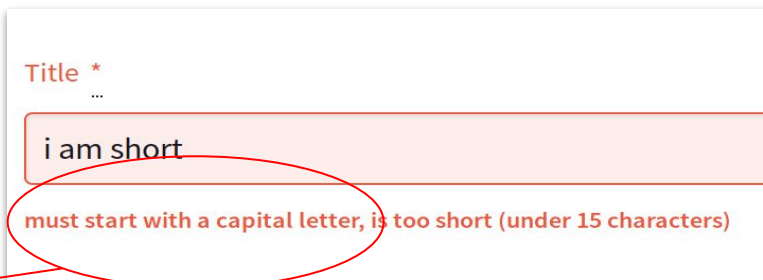
Eve O'Hara

Continue

Step 1: Find the class with the method.

This can be difficult and requires time to study the code to do it properly. But... we're feeling brave!

TIP: start to search for the locale key and then where is it placed in the code:



A screenshot of a GitHub search interface. The search bar contains the text "must start with a capital" and a "Search" button. The results show "1 code result in decidim/decidim" with a link to "view all results on GitHub". The code snippet is from "decidim-core/config/locales/en.yml" and contains the following lines:

```
787     title: Reply
788     show:
789       title: Conversation with %{usernames}
790       start:
791         send: Send
...
1224     long_words: contains words that are too long (over 35 characters)
1225     must_start_with_caps: must start with a capital letter
1226     nesting_too_deep: can't be inside of a subcategory
```

The search bar and the error message "must start with a capital" are circled in red. A red arrow points from the error message in the top right to the search bar. The file path "decidim-core/config/locales/en.yml" and the key "must_start_with_caps" are also circled in red.

Search where the locale key is used

Search

Code 1

Commits 0

Issues 0

Packages 0

Languages

YAML 25

Ruby X

1 code result in [decidim/decidim](#)

[decidim-core/app/validators/etiquette_validator.rb](#)

Bingo!

```
39     record.errors.add(attribute, options[:message] || :must_start_with_caps)
40   end
41
42   def validate_length(record, attribute, value)
```

● Ruby Showing the top match Last indexed on 18 Jun

Step 2: Create a new initializer file. It will be used to monkey-patch the method found.

For instance:

config/initializers/bananas_for_proposals.rb

Also: open the original file to copy and analyze what we need.

```
# This validator takes care of ensuring the validated content is
# respectful, doesn't use caps, and overall is meaningful.
class EtiquetteValidator < ActiveRecord::EachValidator
```

```
def validate_length(record, attribute, value)
  return if value.length > 15

  record.errors.add(attribute, options[:message] || :too_short)
end
```

<https://github.com/decidim/decidim/blob/cc44574a14f7935dc6195de0c43486f69d824830/decidim>

```
2
3 # This validator takes care of ensuring the validated content is
4 # respectful, doesn't use caps, and overall is meaningful.
5 class EtiquetteValidator < ActiveRecord::EachValidator
6   def validate_each(record, attribute, value)
7     return if value.blank?
8
9     validate_caps(record, attribute, value)
10    validate_marks(record, attribute, value)
11    validate_long_words(record, attribute, value)
12    validate_caps_first(record, attribute, value)
13    validate_length(record, attribute, value)
14  end
15
16  private
17
18  def validate_caps(record, attribute, value)
19    return if value.scan(/[A-Z]/).length < value.length / 4
20
21    record.errors.add(attribute, options[:message] || :too_much_caps)
22  end
23
24  def validate_marks(record, attribute, value)
25    return if value.scan(/!\?;:}{2,}/).empty?
26
27    record.errors.add(attribute, options[:message] || :too_many_marks)
28  end
29
30  def validate_long_words(record, attribute, value)
31    return if value.scan(/[A-Z]{35,}/).empty?
32
33    record.errors.add(attribute, options[:message] || :long_words)
34  end
35
36  def validate_caps_first(record, attribute, value)
37    return if value.scan(/\^[a-z]{1}/).empty?
38
39    record.errors.add(attribute, options[:message] || :must_start_with_caps)
40  end
41 end
```

Step 3: Edit the file and register our hook for the class we want to monkeypatch.
Copy the method exactly as it is. Then modify what you want.

```
EXPLORER
OPEN EDITORS
UNTITLED (WORKSP...
> decidim
v decidim-hacks
  > app
  > bin
  v config
    > environments
    v initializers
      application_controller_rendere...
      assets.rb
      backtrace_silencers.rb
      bananas_for_proposals.rb U
      config_overrides

bananas_for_proposals.rb x
decidim-hacks > config > initializers > bananas_for_proposals.rb
1 EtiquetteValidator.class_eval do
2   def validate_length(record, attribute, value)
3     return if value.length > 5
4
5     record.errors.add(attribute, options[:message] || :too_short)
6   end
7 end
```

Step 4: Reload the server. Everything modified inside initializers requires a server reload.

← Back
CREATE YOUR PROPOSAL

You are creating a proposal.

- Create your proposal
- Compare

Title *
...

i am short

must start with a capital letter

Body *
...

 **Behaviour modified!** 

← Back
COMPLETE YOUR PROPOSAL

You are creating a proposal.

- Create your proposal
- Compare
- Complete

Title *
...

I am short

Body *
...

Hello, I want to hack!

Inspire yourself: real examples

<https://omastadi.hel.fi/>

<https://plazapublica.cdmx.gob.mx/>

<https://dddc.decodeproject.eu/>

<https://labs.openheritage.eu/>



<https://github.com/City-of-Helsinki/decidim-helsinki>

<https://github.com/ponentesincausa/PlazaPublicaCDMX>

<https://github.com/DECODEproject/DDDC-instance>

<https://github.com/Platoniq/decidim-openheritage>

 <https://github.com/decidim/decidim/network/dependents>

Conclusions

1. Don't do it!
2. Be aware of upgrades!
3. Touch the minimum possible.
4. Test extensively in local before deploying in production



Leave your comments in

<https://github.com/Platoniq/decidim-hacks/issues>

or just write me:

ivan@platoniq.net



<https://twitter.com/ivanverges>



<https://meta.decidim.org/profiles/microstudi>



<https://github.com/microstudi>



ivan@platoniq.net