



Drupal Europe

Darmstadt, Germany

Sep 10 - 14, 2018

www.drupaleurope.org



Drupal + Technology

TRACK SUPPORTED BY





Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

VUEJS and Drupal

How to build a decoupled SPA

Alvaro J. Hurtado Villegas



Alvaro J. Hurtado Villegas

Drupal Developer at front.id

 @alvar0hurtad0

What's this presentation about?

Explain an example

- General structure for a decoupled application.
- How to build the backend with drupal.
- General structure of a VUEJS application.
- The API part in VUEJS.
- Authentication.
- Interceptors.
- Components in VUEJS.
- The router component.
- How to communicate components in VUEJS (VUEX).
- Vuetify.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

**I head and I forget
I see and I remember
I do, I understand**

Confucious



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

**I completely broke the example and I fixed
I enjoyed learning :)**

http://drupal.org/u/c0nfuc1o_us *

* that's just a joke. This username doesn't exist on D.O.



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

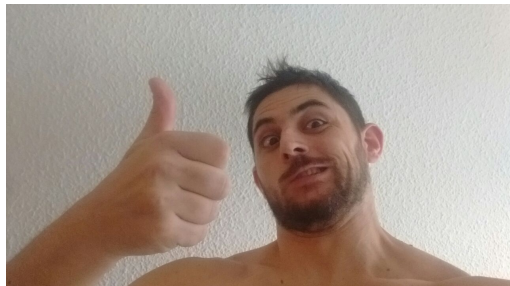
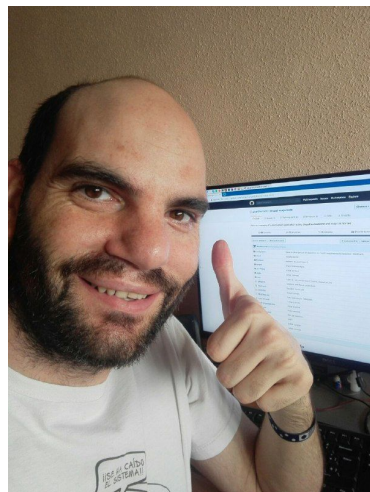
<https://github.com/alvar0hurtad0/drupal-vuejs-todo>

please find this example

Follow README instructions



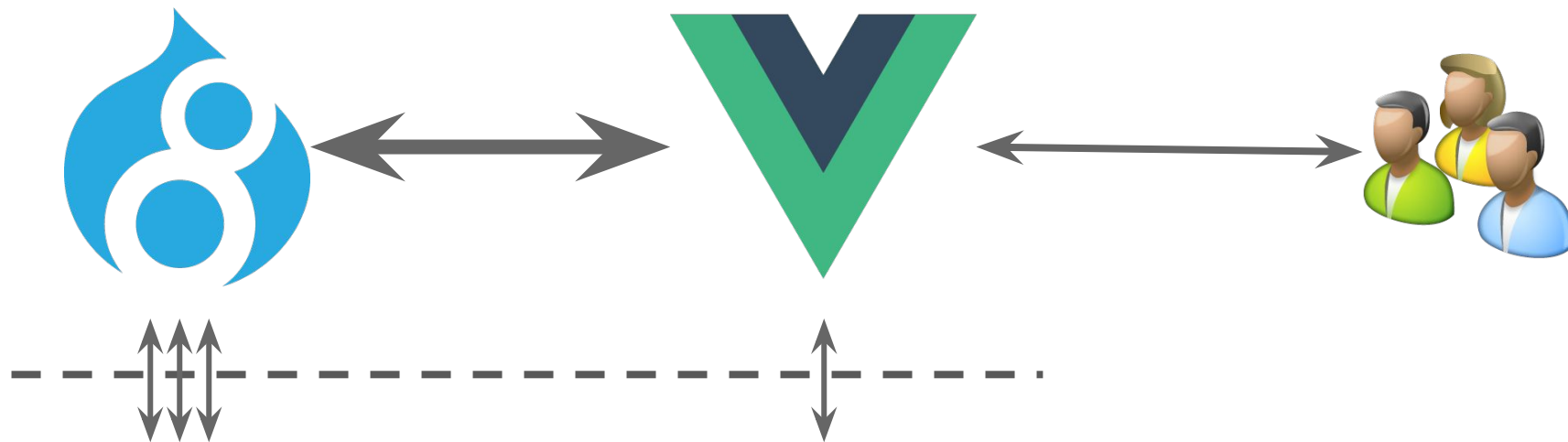
Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



General structure



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Other services to use

Repository overview



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

config/sync

drush

frontend

scripts

src/Phing

tests

web

We can move into another repository



Drupal project with special folder called frontend

Backend with drupal

On the example



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

[drupal-vuejs-todo](#) / [web](#) / [modules](#) / [custom](#) / [dvt_rest_api](#) / [src](#) / [Plugin](#) / [rest](#) / [resource](#) / [TodoResource.php](#)

← Create a plugin to implement a REST resource

 alvar0hurtad0 Fix coding standards.

1 contributor

206 lines (176 sloc) | 5.25 KB

Raw

```
1 <?php
2
3 namespace Drupal\dvt_rest_api\Plugin\rest\resource;
4
5 use Drupal\rest\Plugin\ResourceBase;
6 use Drupal\rest\ResourceResponse;
7 use Drupal\Core\Session\AccountProxyInterface;
8 use Psr\Log\LoggerInterface;
9 use Drupal\Core\Entity\EntityStorageInterface;
10 use Symfony\Component\HttpKernel\Exception\NotFoundHttpException;
11 use Symfony\Component\DependencyInjection\ContainerInterface;
12
13 /**
14  * Provides a Resource to list latest posts.
15  *
16  * @RestResource(
17  *   id = "dvt_todos_resource",
18  *   label = @Translation("Todos resource."),
19  *   uri_paths = {
20  *     "canonical" = "/api/v1/todos/{id}",
21  *     "create" = "/api/v1/todos",
22  *   }
23  * )
24  */
25 class TodoResource extends ResourceBase {
```

← Use annotations to tell drupal the paths

Backend with drupal

On the example



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

[drupal-vuejs-todo](#) / [web](#) / [modules](#) / [custom](#) / [dvt_rest_api](#) / [src](#) / [Plugin](#) / [rest](#) / [resource](#) / [TodoResource.php](#)

← Create a plugin to implement a REST resource

 alvar0hurtad0 Fix coding standards.

1 contributor

206 lines (176 sloc) | 5.25 KB

Raw

```
1 <?php
2
3 namespace Drupal\dvt_rest_api\Plugin\rest\resource;
4
5 use Drupal\rest\Plugin\ResourceBase;
6 use Drupal\rest\ResourceResponse;
7 use Drupal\Core\Session\AccountProxyInterface;
8 use Psr\Log\LoggerInterface;
9 use Drupal\Core\Entity\EntityStorageInterface;
10 use Symfony\Component\HttpKernel\Exception\NotFoundHttpException;
11 use Symfony\Component\DependencyInjection\ContainerInterface;
12
13 /**
14  * Provides a Resource to list latest posts.
15  *
16  * @RestResource(
17  *   id = "dvt_todos_resource",
18  *   label = @Translation("Todos resource."),
19  *   uri_paths = {
20  *     "canonical" = "/api/v1/todos/{id}",
21  *     "create" = "/api/v1/todos",
22  *   }
23  * )
24  */
25 class TodoResource extends ResourceBase {
```

← Use annotations to tell drupal the paths

Backend with drupal

Implement your GET, POST, PATH, DELETE... methods.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

```
public function get($id) {
    $query = $this->nodeStorage->getQuery()
        // Get only published todos.
        ->condition('status', 1)
        ->condition('type', 'todo')
        // Get only current user's todos.
        ->condition('uid', $this->currentUser->id());

    // If we're trying to get a specific todo.
    if ($id != 'all') {
        $query->condition('nid', $id);
    }

    $result = $query->execute();

    // Get nodes.
    $nodes = $this->nodeStorage->loadMultiple($result);

    $response = new ResourceResponse($nodes);

    // Alter caches.
    $response->getCacheableMetadata()
        ->addCacheTags(['dvt_todos']);

    return $response;
}
```

```
public function delete(int $nid) {
    $node = $this->nodeStorage->load($nid);

    // If could not load the entity.
    if (!$node) {
        throw new NotFoundHttpException('Missing entity to remove.');
```

```
    }
    $node->delete();

    $response = new ResourceResponse();

    return $response;
}

public function patch(int $nid, array $payload) {
    // Check payload format.
    if (!isset($payload['title']) || !isset($payload['status'])) {
        throw new NotFoundHttpException('Missing parameters on payload.');
```

```
    }

    $node = $this->nodeStorage->load($nid);

    // If could not load the entity.
    if (!$node) {
        throw new NotFoundHttpException('Missing entity to edit.');
```

```
    }
    // @todo: It'll be nice control permissions before update. I'll not do it
    // to maintain the example as simpler as possible.
    $node->set('title', $payload['title']);
    $node->set('field_done', $payload['status']);

    $node->save();

    $response = new ResourceResponse($node);
    return $response;
}
```

```
public function post(array $payload) {
    // Check payload format.
    if (!isset($payload['title']) || !isset($payload['status'])) {
        throw new NotFoundHttpException('Missing parameters on payload.');
```

```
    }

    // Create the node.
    $node = $this->nodeStorage->create([
        'type' => 'todo',
        'title' => $payload['title'],
        'field_done' => $payload['status'],
    ]);
    $node->save();

    // Build a response with the created node.
    return new ResourceResponse($node);
}
```


Backend with drupal

You'll need also a config entity



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

[drupal-vuejs-todo](#) / [web](#) / [modules](#) / [custom](#) / [dvt_rest_api](#) / [config](#) / [install](#) / [rest.resource.dvt_todos_resource.yml](#)

 [alvar0hurtad0](#) Update rest.resource.dvt_todos_resource.yml

1 contributor

23 lines (22 sloc) | 334 Bytes

Raw

Blame

```
1 langcode: en
2 status: true
3 dependencies:
4   module:
5     - basic_auth
6     - dvt_rest_api
7     - serialization
8     - user
9 id: dvt_todos_resource
10 plugin_id: dvt_todos_resource
11 granularity: resource
12 configuration:
13   methods:
14     - GET
15     - POST
16     - DELETE
17     - PATCH
18   formats:
19     - json
20   authentication:
21     - basic_auth
22     - cookie
```



Don't forget about allowed methods, formats and authentication.

Backend with drupal

The resource has its own permissions



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER
RESTful Web Services		
Access DELETE on <i>Todos resource</i> . resource	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Access GET on <i>Todos resource</i> . resource	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Access PATCH on <i>Todos resource</i> . resource	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Access POST on <i>Todos resource</i> . resource	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Administer REST resource configuration	<input type="checkbox"/>	<input type="checkbox"/>

Backend with drupal

Remember to clear the cache



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾ [drupal-vuejs-todo](#) / [web](#) / [modules](#) / [custom](#) / [dvt_rest_api](#) / [dvt_rest_api.module](#)

```
10  /**
11   * Implements hook_node_presave().
12   *
13   * Invalidates todos cache.
14   */
15  function dvt_rest_api_node_presave(Node $node) {
16    // Invalidates todos cache after any changes.
17    if ($node->bundle() == 'todo') {
18      \Drupal::service('cache_tags.invalidator')->invalidateTags(['dvt_todos']);
19    }
20  }
21
22  /**
23   * Implements hook_node_delete().
24   *
25   * Invalidates todos cache.
26   */
27  function dvt_rest_api_node_delete(Node $node) {
28    // Invalidates todos cache after any changes.
29    if ($node->bundle() == 'todo') {
30      \Drupal::service('cache_tags.invalidator')->invalidateTags(['dvt_todos']);
31    }
32  }
```

Backend with drupal

CORS



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾

drupal-vuejs-todo / web / sites / default / services.yml

```
161 cors.config:
162   enabled: true
163   # Specify allowed headers, like 'x-allowed-header'.
164   allowedHeaders: ['*']
165   # Specify allowed request methods, specify '*' to allow all possible ones.
166   allowedMethods: ['*']
167   # Configure requests allowed from specific origins.
168   allowedOrigins: ['*']
169   # Sets the Access-Control-Expose-Headers header.
170   exposedHeaders: false
171   # Sets the Access-Control-Max-Age header.
172   maxAge: false
173   # Sets the Access-Control-Allow-Credentials header.
174   supportsCredentials: true
```

Should be more restrictive
for live environments.

Alternatives

JSONAPI

- Makes things easier on the backend side
- Easier documentation (swagger)
- More flexible API
 - Queries
 - Alter the structure
- More work on the frontend side
- Sometimes you need to do a lot of requests (subrequests module)



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

THE FRONTEND

PART

Overview



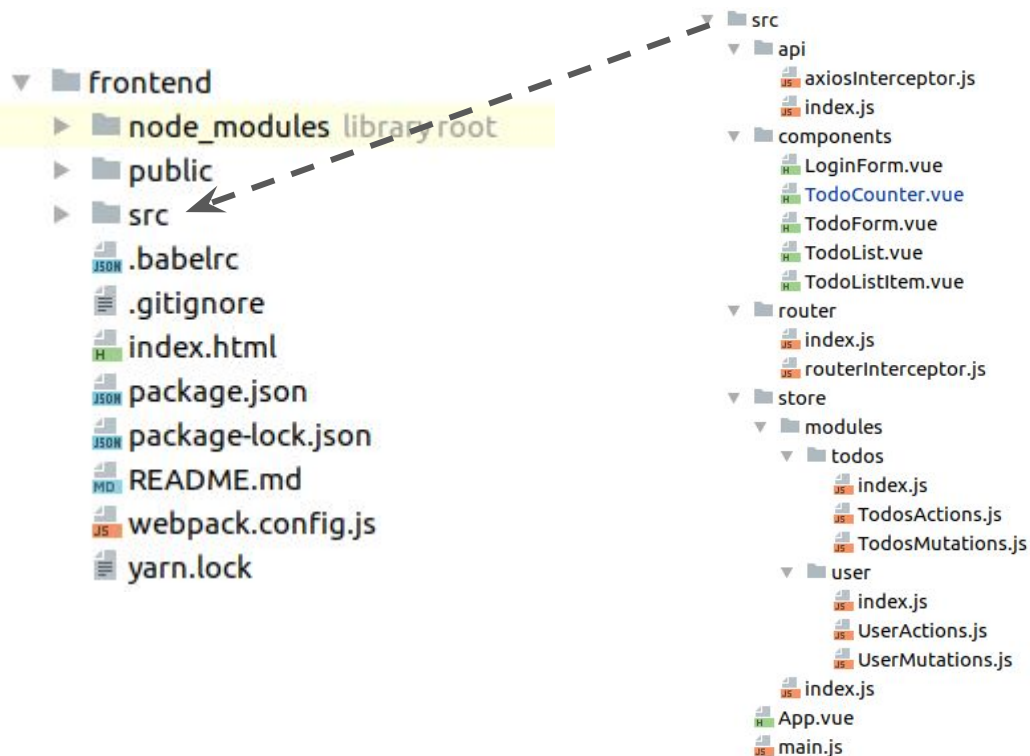
Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

The screenshot shows a web browser window with the address bar displaying 'localhost:8082/#/...'. The browser's taskbar at the top lists several applications: 'Aplicaciones', 'startdebugger', 'stopdebugger', 'drupal', and 'Otros marcadores'. The main content area of the browser shows a login form titled 'Login'. The form contains two input fields: 'username*' and 'Password*', both with asterisks indicating they are required. A mouse cursor is positioned over the 'username*' field. Below the password field is a blue 'LOGIN' button. At the bottom of the browser window, a status bar displays the text 'drupal&vuejs rock!!!'.

Overview



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



The API



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾

[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [api](#) / [index.js](#)

```
addTodo(payload) {  
  return axios.post(`${BASE_API_URL}/api/v1/todos?_format=json`, {title:payload.title, status: false})  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},  
  
editTodo({id, title, status}) {  
  return axios.patch(`${BASE_API_URL}/api/v1/todos/${id}?_format=json`, {title: title, status: status})  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},  
  
removeTodo(id) {  
  return axios.delete(`${BASE_API_URL}/api/v1/todos/${id}?_format=json`)  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},  
  
getTodos() {  
  return axios.get(`${BASE_API_URL}/api/v1/todos/all?_format=json`)  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},
```

Authentication



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾

[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [api](#) / [index.js](#)

```
getToken(payload) {  
  let data = JSON.stringify({  
    name: payload.username,  
    pass: payload.password  
  });  
  
  let config = JSON.stringify({  
    headers: {  
      "Content-Type": "application/json"  
    }  
  });  
  
  localStorage.clear();  
  
  return axios.post(`${BASE_API_URL}/user/login?format=json`, data, config)  
    .then(res => {  
      localStorage.setItem('loggedIn', true);  
      localStorage.setItem('csrf_token', res.data.csrf_token);  
      localStorage.setItem('username', payload.username);  
      localStorage.setItem('auth_token', btoa(payload.username + ':' + payload.password));  
      // @todo: This redirect should be done by vue-router.  
      window.location.href = '/';  
      Promise.resolve(res.data);  
    })  
    .catch(({ error }) => {  
      localStorage.setItem('loggedIn', false);  
      console.log('Issue with login');  
      Promise.reject(error);  
    })  
  },
```

Basic auth is not a very secure authentication method but it's easy to implement for an example application.

← — It just stored all required data on localstorage.
The weakest part of this is store username and password encoded in base64.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

How are the tokens injected?

Remember this:

```
addTodo(payload) {  
  return axios.post(`${BASE_API_URL}/api/v1/todos?_format=json`, {title:payload.title, status: false})  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},  
  
editTodo({id, title, status}) {  
  return axios.patch(`${BASE_API_URL}/api/v1/todos/${id}?_format=json`, {title: title, status: status})  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},  
  
removeTodo(id) {  
  return axios.delete(`${BASE_API_URL}/api/v1/todos/${id}?_format=json`)  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},  
  
getTodos() {  
  return axios.get(`${BASE_API_URL}/api/v1/todos/all?_format=json`)  
    .then((response) => Promise.resolve(response.data))  
    .catch((error) => Promise.reject(error));  
},
```

Interceptors



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▼

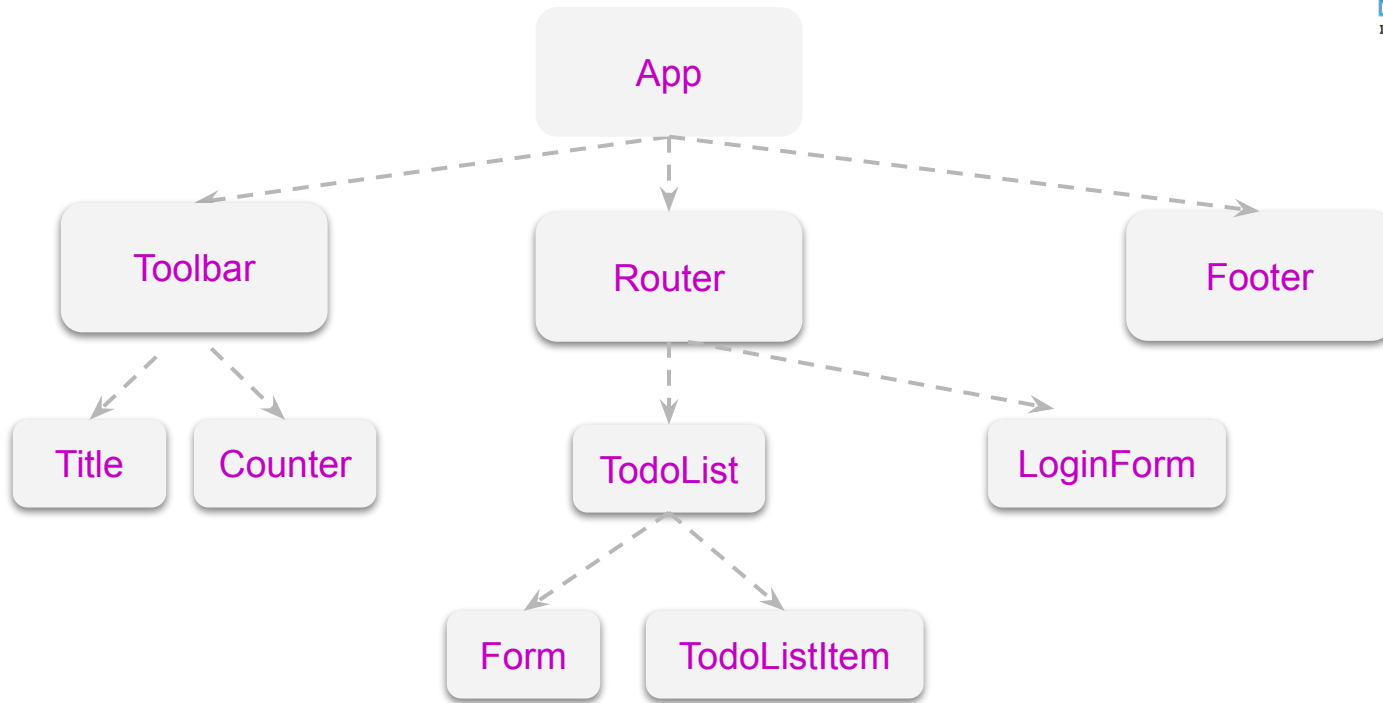
[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [api](#) / [axiosInterceptor.js](#)

```
axios.interceptors.request.use(function(config) {  
  const authToken = localStorage.getItem('auth_token');  
  const csrfToken = localStorage.getItem('csrf_token');  
  if(authToken && csrfToken) {  
    config.headers.common = {  
      "Authorization": `Basic ${authToken}`,  
      "CSRF-Token": csrfToken,  
      "Accept": "application/json, text/plain, */*",  
      "Content-Type": "application/json"  
    };  
  }  
  return config;  
}, function(err) {  
  return Promise.reject(err);  
});
```

Components



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Components



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

```
<template>
  <v-app>
    <v-toolbar app fixed>
      <v-toolbar-title v-text="title"></v-toolbar-title>
      <v-spacer></v-spacer>
      <todo-counter v-if="User.isLoggedIn"></todo-counter>
      <v-btn to="Logout" fab small v-if="User.isLoggedIn">
        <v-icon>logout</v-icon>
      </v-btn>
    </v-toolbar>
    <v-content>
      <v-container fluid>
        <router-view></router-view>
      </v-container>
    </v-content>
    <v-footer :fixed="true" app>
      <span>drupal&vuejs rock!!!</span>
    </v-footer>
  </v-app>
</template>
```

```
<script>
  import TodoCounter from './components/TodoCounter';
  import { mapActions } from 'vuex';
  import { mapState } from 'vuex';

  export default {
    components: {
      'todo-counter': TodoCounter,
    },
  },
}
```

There're also components provided by libraries.

And of course HTML

Components



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: **master** ▾

[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [components](#) / **TodoList.vue**

```
<template>
  <v-layout row wrap>
    <v-flex xs12>
      <todo-form></todo-form>
    </v-flex>
    <v-flex xs12>
      <v-list>
        <todo-list-item v-for="item in items" :key="item.id" v-bind:id="item.id" />
      </v-list>
    </v-flex>
  </v-layout>
</template>
```

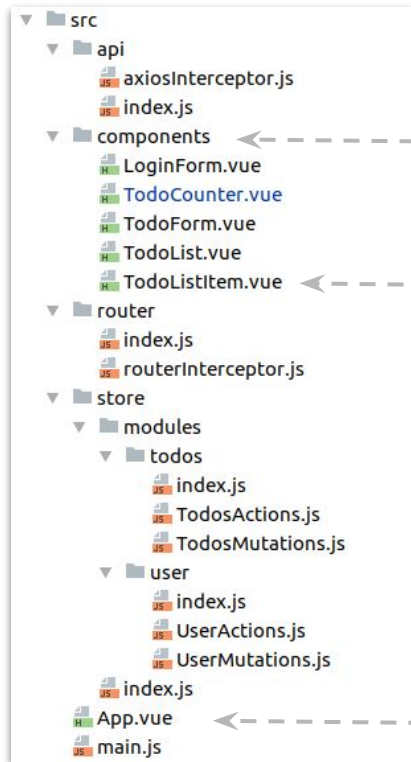
```
<script>
  import { mapActions, mapGetters } from 'vuex';
  import TodoListItem from './TodoListItem';
  import TodoForm from './TodoForm';

  export default {
    components: {
      TodoForm,
      'todo-list-item': TodoListItem,
      'todo-form': TodoForm,
    },
    methods: {
      ...mapActions(['getTodos', 'getToken']),
    },
    computed: {
      items() {
        return this.$store.state.Todos.todos;
      }
    },
    mounted() {
      this.getTodos();
    },
  }
</script>
```

Components



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



All sub components.
Can be reorganize in folders.

If a component only makes sense into another. We Should prefix the name with parent's name.

Our main component

Vuetify components



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

```
<template>
  <v-app>
    <v-toolbar app fixed>
      <v-toolbar-title v-text="title"></v-toolbar-title>
      <v-spacer></v-spacer>
      <todo-counter v-if="User.isLoggedIn"></todo-counter>
      <v-btn to="Logout" fab small v-if="User.isLoggedIn">
        <v-icon>logout</v-icon>
      </v-btn>
    </v-toolbar>
    <v-content>
      <v-container fluid>
        <router-view></router-view>
      </v-container>
    </v-content>
    <v-footer :fixed="true" app>
      <span>drupal&vuejs rock!!!</span>
    </v-footer>
  </v-app>
</template>
```

And many more ...

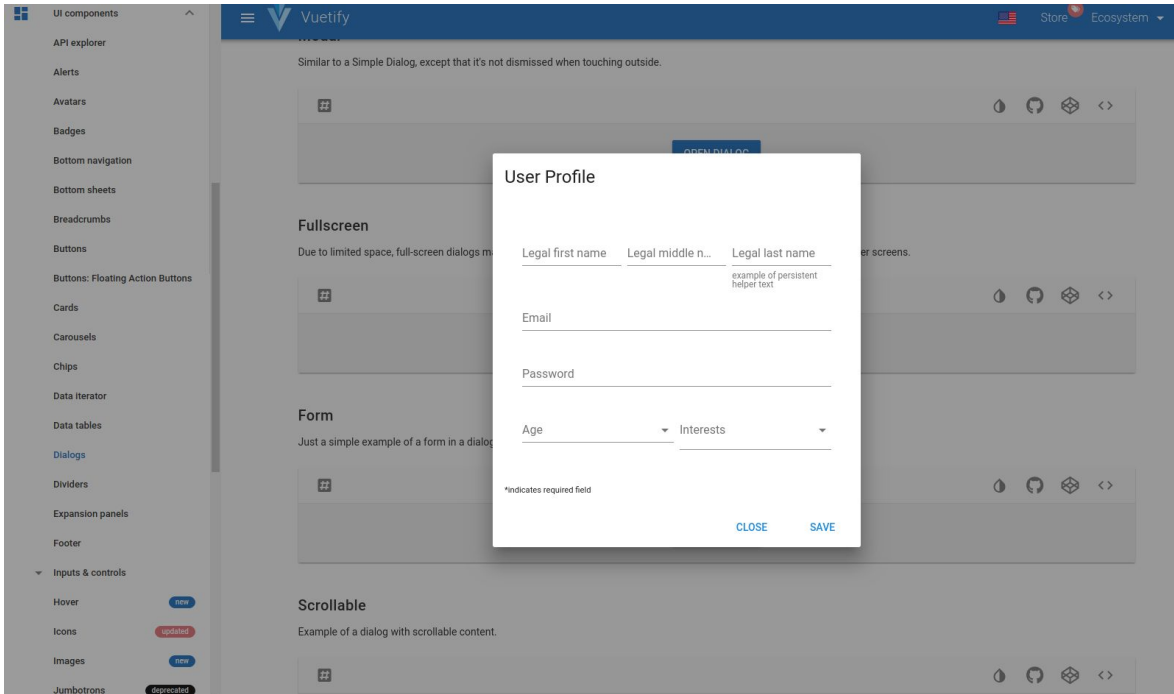
- Have a lot of configuration options.
- Can inject color schemes.
- More common components for an material design app.
- Complete responsive layout system Out of the box.

Vuetify website

<https://vuetifyjs.com>



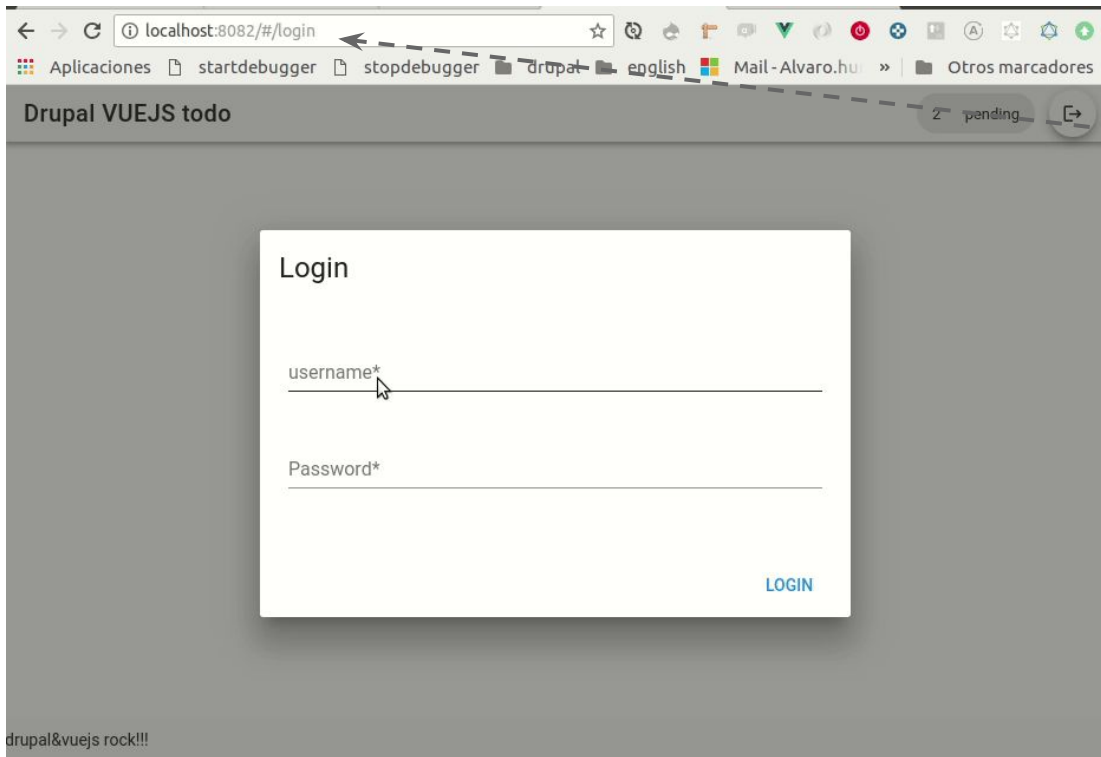
Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Router



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Look at this

drupal&vuejs rock!!!

Router



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾

drupal-vuejs-todo / frontend / src / router / index.js

```
import Vue from 'vue'
import Router from 'vue-router'
import TodoList from '@/components/TodoList'
import LoginForm from '@/components/LoginForm'

Vue.use(Router);

export default new Router({
  routes: [
    {
      path: '/login',
      name: 'Login',
      component: LoginForm
    },
    {
      path: '/logout',
      name: 'Logout',
    },
    {
      path: '/',
      name: 'List',
      component: TodoList,
      meta: {
        requiresAuth: true
      },
    },
  ],
})
```

Match paths with
components

We can extend the data
structure.

Router

Branch: **master** ▾

[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [router](#) / [routerInterceptor.js](#)



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

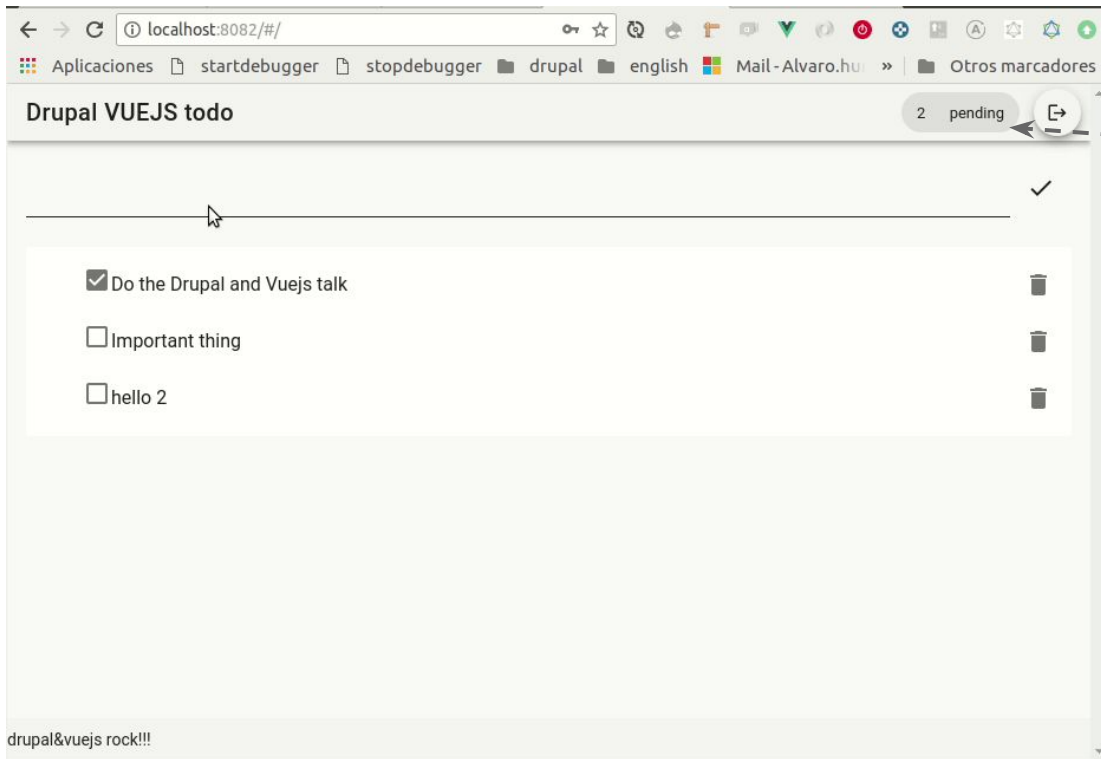
```
router.beforeEach((to, from, next) => {
  // If visiting login view but you already have logged in redirect to home.
  if (to.name === 'Login' && window.localStorage.loggedIn) {
    return next({
      path: '/'
    })
  }

  // Remove session on logout.
  if (to.name === 'Logout') {
    delete window.localStorage.loggedIn;
    return next({
      path: '/login'
    })
  }

  // Redirect to login all request that need authentication if user is not logged.
  if(to.meta.requiresAuth && !(window.localStorage.loggedIn)){
    return next({
      path: '/login',
      query: {
        redirect: to.fullPath
      }
    })
  }

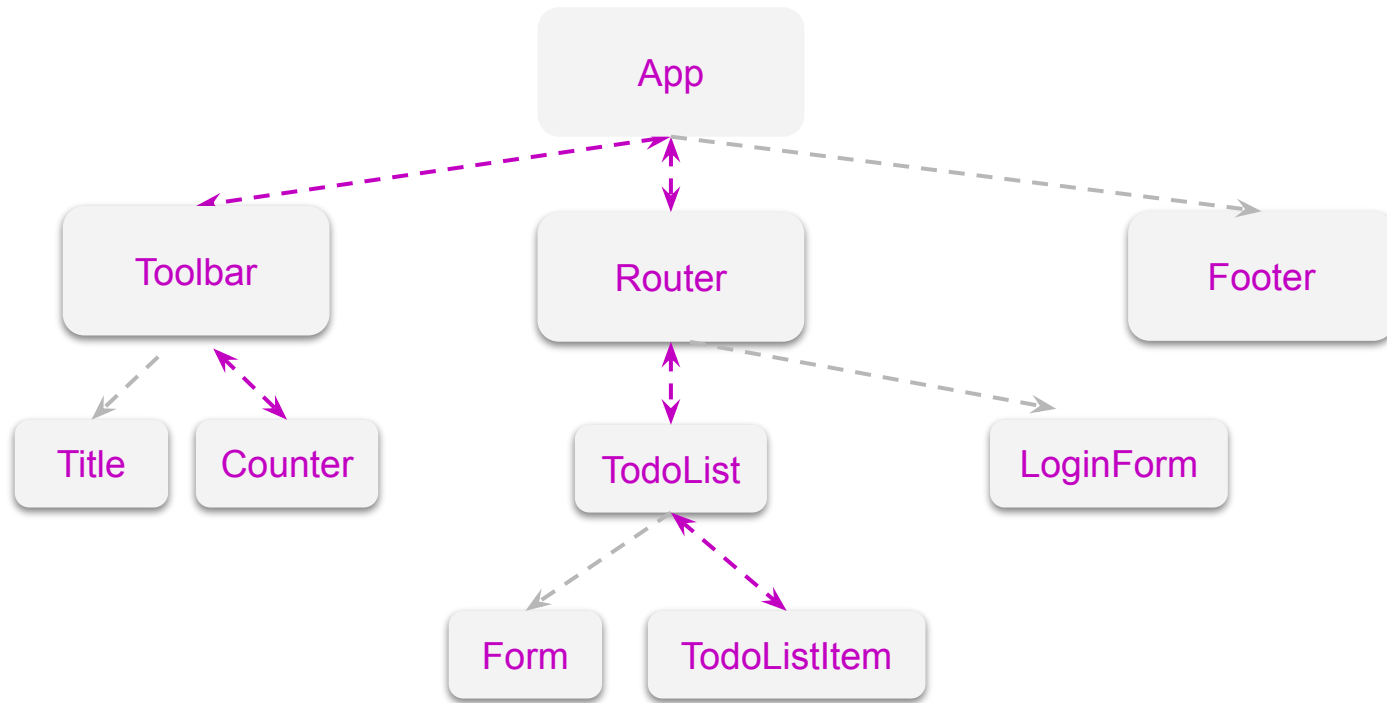
  return next()
})
```

Components communication



Look at this

I click on a todo and the counter reacts



Vuex



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾

drupal-vuejs-todo / frontend / src / store / index.js

```
import Vue from 'vue'
import Vuex from 'vuex';
import Todos from '../modules/todos';
import User from '../modules/user';

const actions = {};
const getters = {};
const mutations = {};
const state = {};

Vue.use(Vuex);

export default new Vuex.Store({
  actions,
  getters,
  modules: {
    Todos,
    User
  },
  mutations,
  state
});
```

Logic and async operations.

Complex queries to reuse.

The only place to change the state

All the data.

Vuex



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: master ▾

[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [store](#) / [modules](#) / [todos](#) / **TodosActions.js**

```
import actions from './TodosActions';
import mutations from './TodosMutations';

export default ({
  namespaced: true,
  state: {
    todos: [], ← ----- Initial state.
  },
  actions,
  mutations,
});
```

Vuex



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

```
import {ADD_TODO, REMOVE_TODO, SET_TODOS, TOGGLE_TODO} from "../TodosMutations";
import api from "@api"
export default {
  getTodos({commit}, payload) {
    api.getTodos()
      .then((response) => commit(SET_TODOS, response))
      .catch((error) => { console.log('Error: ', error) });
  },
  addTodo({commit}, payload) {
    api.addTodo(payload);
    commit(ADD_TODO, payload)
  },
  removeTodo({commit}, payload) {
    api.removeTodo(payload);
    commit(REMOVE_TODO, payload)
  },
  toggleTodo({commit}, payload) {
    api.editTodo(payload);
    commit(TOGGLE_TODO, payload)
  }
}
```

Actions

```
export default {
  [ADD_TODO](state, { id, title }) {
    state.todos.push({ id, title, status: false })
  },
  [REMOVE_TODO](state, id) {
    var index = -1;
    for(var i = 0; i < state.todos.length; i++) {
      if(state.todos[i].id == id) {
        index = i;
      }
    }

    if (index != -1) {
      state.todos.splice(index, 1);
    }
  },
}
```

Mutations

Vuex



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Branch: **master** ▾

[drupal-vuejs-todo](#) / [frontend](#) / [src](#) / [components](#) / **TodoList.vue**

```
<template>
  <v-layout row wrap>
    <v-flex xs12>
      <todo-form></todo-form>
    </v-flex>
    <v-flex xs12>
      <v-list>
        <todo-list-item v-for="item in items" :key="item.id" v-bind:id="item.id" />
      </v-list>
    </v-flex>
  </v-layout>
</template>
```

```
<script>
  import { mapActions } from 'vuex';
  import TodoListItem from './TodoListItem';
  import TodoForm from './TodoForm';

  export default {
    components: {
      TodoForm,
      'todo-list-item': TodoListItem,
      'todo-form': TodoForm,
    },
    methods: {
      ...mapActions('Todos', ['getTodos']),
    },
    computed: {
      items() {
        return this.$store.state.Todos.todos;
      }
    },
    mounted() {
      this.getTodos();
    },
  }
</script>
```



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

Questions ?



Alvaro J. Hurtado Villegas

Drupal Developer at [front.id](#)

 [@alvar0hurtad0](#)

THANKS !