

Event Management App

Agenda

- Customer Requirements
- Using DataObjects
- Symfony Forms with dynamic constraints
- User-Service using Dependency Injection
- Custom API for native APP
- Extending Pimcore Backend (AdminController + ExtJS)

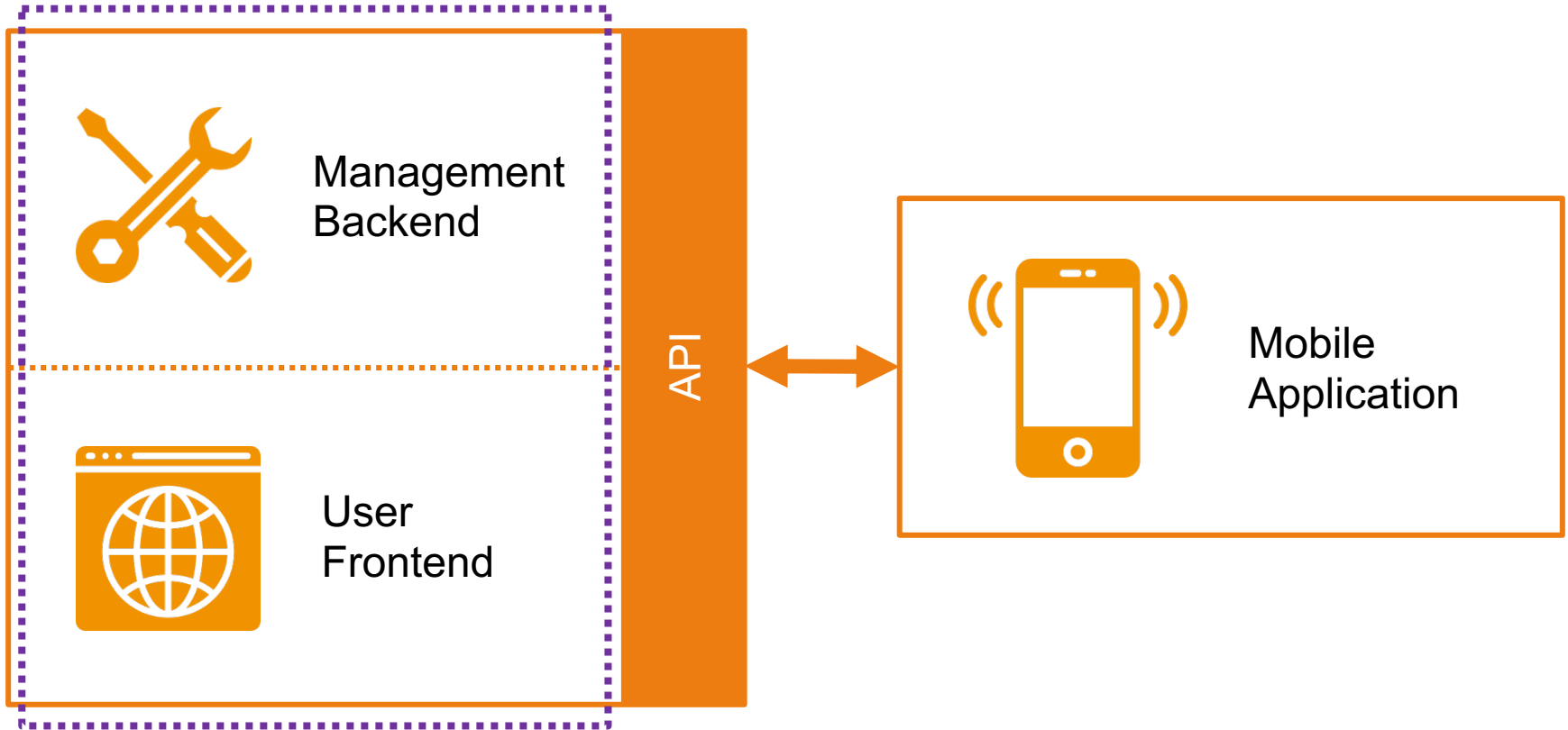


Customer Requirements

- Manage company internal events
- Make participant management more simple
- Provide event custom course overview
- Simple helpdesk
- Useful Backend



PIMCORE®

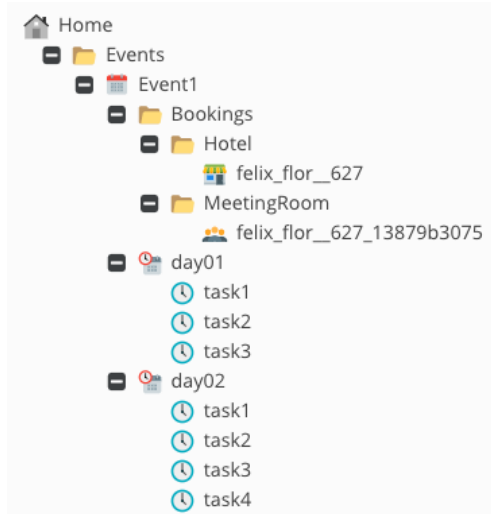


Using DataObjects

- Create and Manage Events
 - Tasks
 - Bookings
- Participant Management
- Push-Notifications
- Form-Options



Event



Event

Basic Informations Contents Downloads Schedule User

Registration is closed:

Start Date *: 2019-05-07 End Date *: 2019-05-10

Name *: Rhenus Management Forum 2019

Address Name: Hotel "Kameha Grand"

Address *: Am Bonner Bogen 1

ZIP *: 53227 City *: Bonn

Country *: Germany

Hotel Booking Folder *: /Events/Event1/Bookings/Hotel

Meeting Room Booking Folder *: /Events/Event1/Bookings/MeetingRoom

Map



Event

Event

Basic Informations

Contents

Downloads

Schedule

User

Days

Date	Day Number	Schedule
2019-05-05	1	/Events/Event1/day01/task1 /Events/Event1/day01/task2 /Events/Event1/day01/task3
2019-05-06	2	/Events/Event1/day02/task1 /Events/Event1/day02/task2 /Events/Event1/day02/task3 /Events/Event1/day02/task4



Event

Event

Basic Informations

Contents

Downloads

Schedule

User

 User

ID	Firstname	Lastname	Email	Member of Board	Contact Language	Account status	Participate
531				<input type="checkbox"/>	English, German	Active	<input checked="" type="checkbox"/>
532				<input type="checkbox"/>	English	Active	<input checked="" type="checkbox"/>
533				<input type="checkbox"/>	German, English	Active	<input checked="" type="checkbox"/>
534				<input type="checkbox"/>		Profile created	<input type="checkbox"/>
535				<input type="checkbox"/>		Profile created	<input type="checkbox"/>
536				<input type="checkbox"/>		Profile created	<input type="checkbox"/>
537				<input type="checkbox"/>		Profile created	<input type="checkbox"/>



User

User

Data User-Image Authentication & Security Billing Informations Further comments

Title: (Empty) ▼

Gender: Male ▼

Firstname *: Felix

Lastname *: Flor

Email *: f.flor@anymotion.de

Function: Developer

Department: Pimcore

Business Area: (Empty) ▼

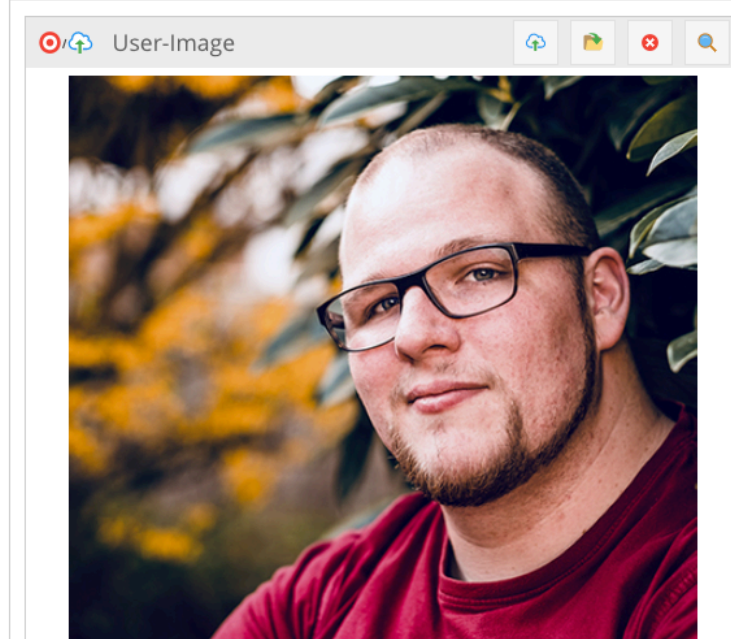
Company: AnyMotion



User

User

Data **User-Image** Authentication & Security Billing Informations Further comments



User

Data User-Image **Authentication & Security** Billing Informations Further comments

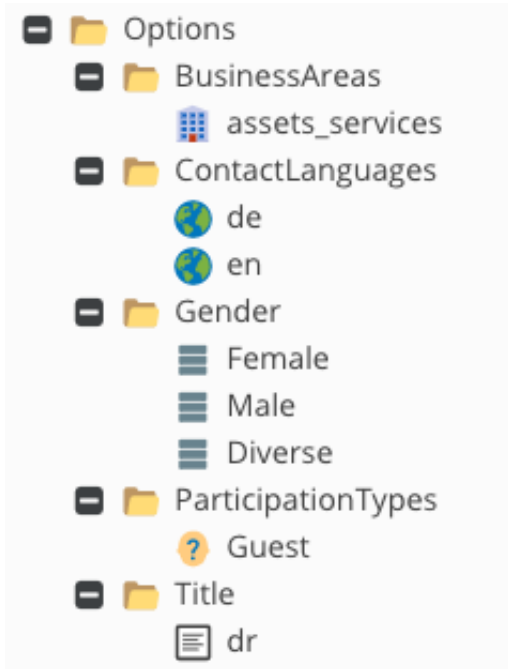
Initial Password:

Password:

Account status:



Form Options / Dynamic Options




- Used in
 - Frontend Forms
 - DataObject Select Types
- Provider Class
- DataObjects to store options



Symfony Forms with dynamic constraints

Profil Image -



Delete Image on next save
 No

Select Image file
 IMG_0251.jpg

Agree to image usage
If an image has been selected, the terms of use must be confirmed.
 No



```
->add( child: 'imageUpload', type: FileType::class, [
    'label' => 'Select Image file',
    'required' => false,
    'mapped' => false,
    'constraints' => [
        new File([
            'groups' => ['userimage'],
            'maxSize' => '8Mi'
        ]),
        new Image([
            'groups' => ['userimage'],
        ])
    ]
])

->add( child: 'agreedToImageUsage', type: CheckboxType::class, [
    'label' => 'Agree to image usage',
    'required' => false,
    'attr' => [
        'switch' => true,
        'id' => 'agreedToImageUsage'
    ],
    'constraints' => [
        new IsTrue([
            'groups' => ['userimage'],
            'message' => 'If an image has been selected, the terms of use must be confirmed.'
        ])
    ]
])
])
```



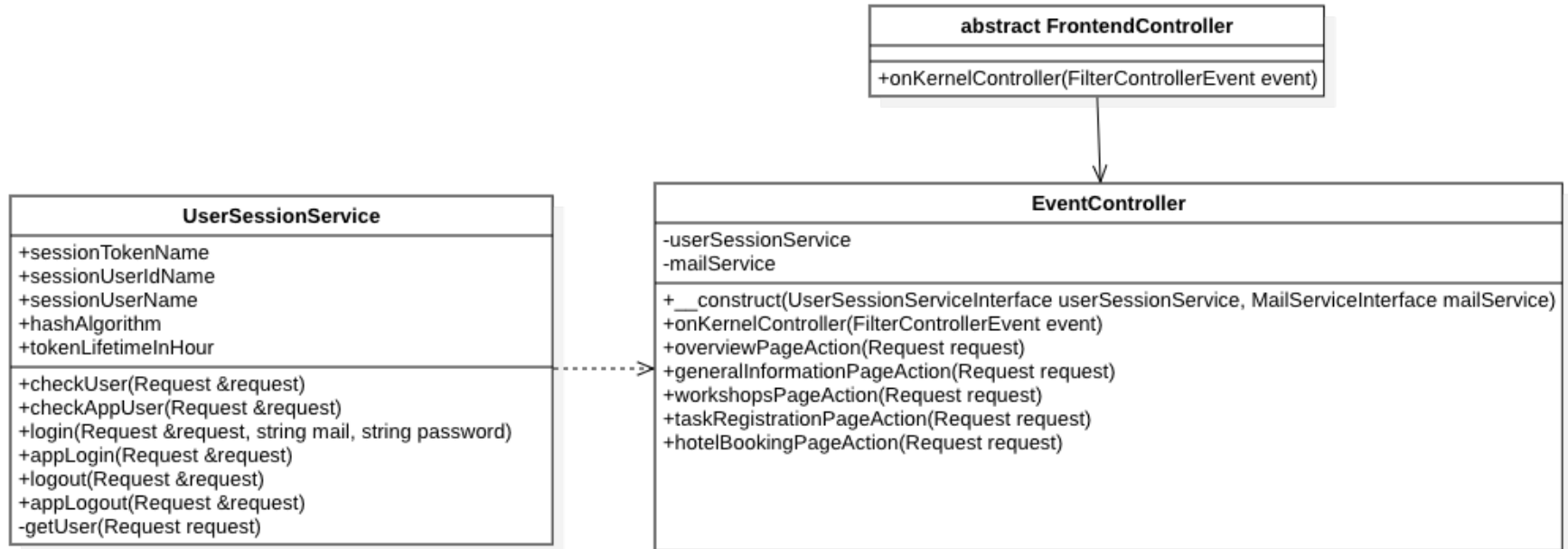
```
...../**
..... * @param OptionsResolver $resolver
..... */
..... public function configureOptions(OptionsResolver $resolver) {
.....     $resolver->setDefaults([
.....         'validation_groups' => function (FormInterface $form){
.....             $groups = ['Default'];
.....             $data = $form->getData();
.....
.....             if($data->get('sameAsUserAddress')){
.....                 $groups[] = 'differentBilling';
.....             }
.....
.....             if(!empty($form['imageUpload']->getData())){
.....                 $groups[] = 'userimage';
.....             }
.....
.....             return $groups;
.....         }
.....     ]);
..... }
```



User-Session-Service

- Handles user login/logout
- Checks user state on request
- Used on Website and App






```

.../**
... * Extends the Parent onKernelController Function (FrontendController in Bundle)
... * for event specific
... *
... * @param FilterControllerEvent $event
... */
... public function onKernelController(FilterControllerEvent $event) {
...     parent::onKernelController($event);
...
...     $request = $event->getRequest();
...
...     if(true !== $redirect = $this->userService->checkUser( &: $request))
...         return $event->setController(function () use ($redirect) { return $redirect; });
...
...     $eventId = $request->get( key: 'eventId');
...     $user = $request->getSession()->get( name: 'user');
...     $userEvent = Event::getById($eventId, force: true);
...
...     $this->userEvent = $userEvent;
...     $this->user = $user;
...
...     if(empty($userEvent))
...         return $event->setController(function () { return new RedirectResponse( url: '/'); });
...
...     if(!$this->userService->isUserParticipant($userEvent, $user)){
...         return $event->setController(function () { return new RedirectResponse( url: '/'); });
...     }
...
...     $this->view->event = $userEvent;
...     $this->view->userHasParticipationConfirmed = $this->userService->userHasParticipationConfirmed($userEvent, $user);
... }

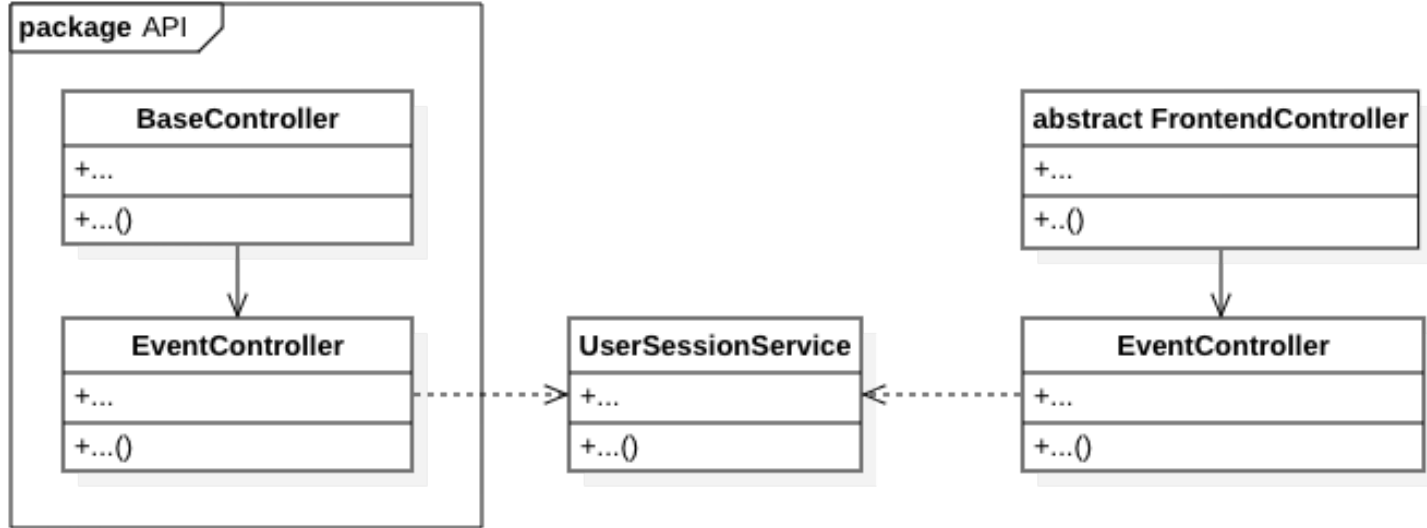
```



Custom API

- Connect native mobile App
- Returns JSON Result
- Requires App-Token and User-Token
- Use the same service classes as app controller classes





\Bundle\Controller\Api

*Uses Annotation
Routes*

\Bundle\Controller

*Uses Pimcore Static
Routes*

Extending Pimcore Backend

- Used to Send and List Push-Notification
- Import Users from “internal System” to Pimcore
- Use ExtJS for integration
- Class extends `\Pimcore\Bundle\AdminBundle\Controller\AdminController`



Documents

- Home
 - event-app-website
 - login
 - footer
 - imprint
 - mail
 - contact
 - reset-password
 - snippets
 - event1

Send Push Notification

- List send Notifications
- Data Import

Send Notification

Send Push Notification

Event: Rhenus Management Forum 2019 (07.05.2019 - 10.05)

Title: My Title

Message: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Pretium quam vulputate dignissim suspendisse in. Nisi lacus sed viverra tellus. Fermentum odio eu feugiat pretium nibh ipsum consequat nisl. Viverra nibh cras pulvinar mattis. Tincidunt dui ut ornare lectus sit. Rhoncus aenean vel elit scelerisque mauris pellentesque. Enim ut tellus elementum sagittis vitae et leo duis. Et malesuada fames ac turpis egestas. Scelerisque viverra mauris in aliquam sem fringilla ut. Nisi pretium fusce id velit ut tortor. Libero nunc consequat interdum varius sit amet mattis vulputate enim.

Send Message to Event-Users



The screenshot shows the PIMCORE interface. On the left is a sidebar with navigation icons and a menu. The menu items are: Documents, Assets, Data Objects, Home, Events, User, Options, Notifications, 12_1567357768 (highlighted), Send Push Notification, List send Notifications, and Data Import. The main content area is titled 'Send Messages' and contains a table with the following data:

Send at	Event	Message Title	Message
01.09.2019 19:09:28	Rhenus Management Forum 2019	My Title	Lorem ipsum do



Thank you for listening.

anyMOTION GRAPHICS GmbH

Malkastenstr. 2

40211 Düsseldorf

Tel.: +49 211 443343

Fax: +49 211 443394

<http://www.anyMOTION.de>

Sämtliche Urheber- und Nutzungsrechte an den in dieser Publikation enthaltenen Arbeiten, Bild- und Schriftwerken einschließlich der zugrunde liegenden Ideen, Schöpfungen und Konzeptionen liegen bei der anyMOTION Graphics GmbH (Düsseldorf). Jede vollumfängliche oder teilweise Reproduktion, Weitergabe oder sonstige Nutzung bedarf unserer vorherigen Zustimmung und ist gesondert honorarpflichtig.

Düsseldorf 2019

