Sport calendar

Idea of sport calendar – is a web application, which helps athletes to track their progress in gym.

When athlete finishes an exercise – he came to our site to save the information. For example, he writes that he did press of a bar 8 times with weight 100kg. This data together with timestamp is saved to database.

Then, on a dashboard athlete sees:

- what exercises he did today,
- what he did a week ago (on the same weekday),
- what he did 2 weeks ago?

In current task we will create only dashboard. There will be fixtures that add data for testing purpose, so we do not need any form to add exercises.

So, go throw steps:

1. **Create new Symfony 3 project on GitHub**

2. **Create data model and fill it with fake data.**

   There is only one entity - Exercise. Fields are: short description of done exercise, weight, count of times exercise was done, date, time (here it's better to store date without time, it'll simplify logic later).
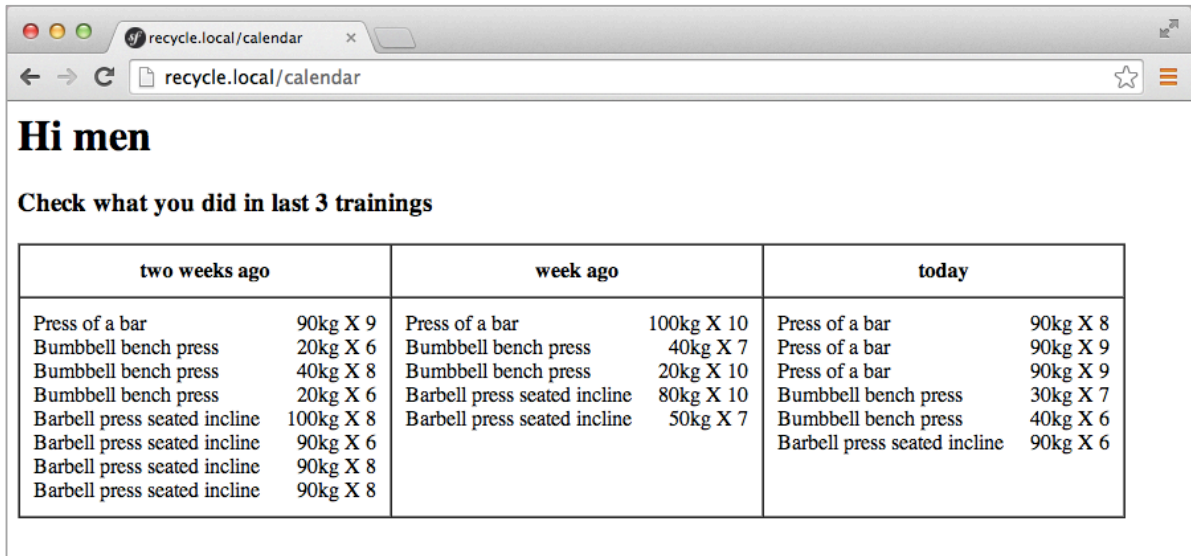
   Tip: You could use build-in code generators to generate a bundle and an entity.

   Then you have to fill data model with fake data (for testing). There is excellent tool for this – Nelmio Alice bundle. Create following fixtures: 3 types of exercises (3 different short descriptions) are randomly imploded into dates of last 30 days, including today. Weight is random between 20 and 200kg. Count of times – is also random, 5-15. Total amount of exercises done in last 30 days should be about 300-400.

   Tip: You may use faker functions like dateTime(),dateTimeBetween(),randomElement(), etc..

3. Implement Front-end

Dashboard should be very simple – just some title and table with 3 columns:

If this project is success, it will require a mobile application with the same dashboard. So, all data should be fetched via service layer, to be reused later.

Tip: Try to implement this functionality without writing custom query
Tip: It's expected to have only one DB request for the whole page

4. Cover Service class with Unit test

Service class is suggested to have one public method to fetch all data, needed for dashboard. This method should be covered with Unit test. Unit test means that all external calls from the service are replaced by mocks. And service class is only one real object in the test, created via "new" operator.

5. Add multi-lingual support of user interface

Translate user interface into 2 languages: Russian and English.
Locale should be defined by part of requested URL. For example, /en/calendar – is English dashboard, /ru/calendar – is Russian one.
Tip: Do not translate data, only interface. So, "press of a bar" – will be the same in both locales, but "today" would change.

What to read before?
https://symfony.com/doc/current/bundles/DoctrineFixturesBundle/index.html#writing-simple-fixtures
https://github.com/nelmio/alice
https://phpunit.de/manual/current/en/test-doubles.html#test-doubles.mock-objects
http://symfony.com/doc/current/book/service_container.html
http://symfony.com/doc/current/book/routing.html
http://symfony.com/doc/current/book/security.html