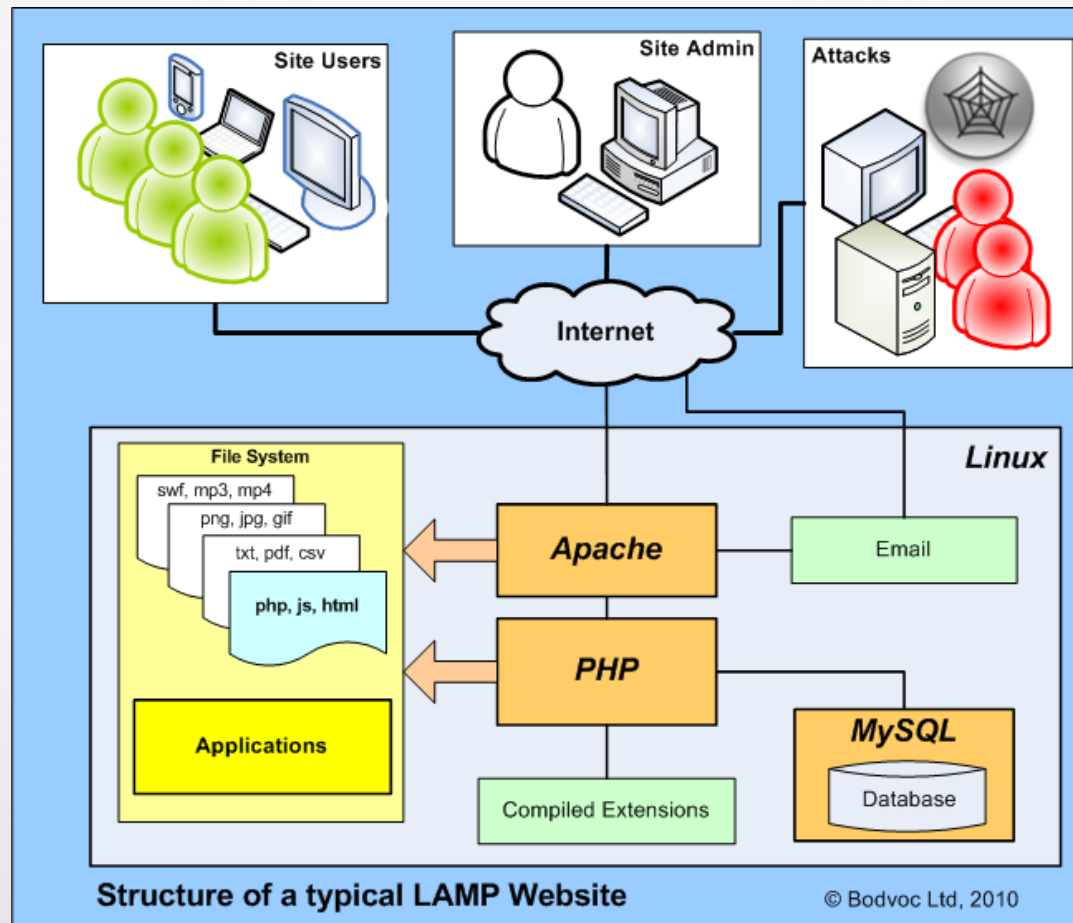


REST API

Alessandro Bugatti

Web application architecture

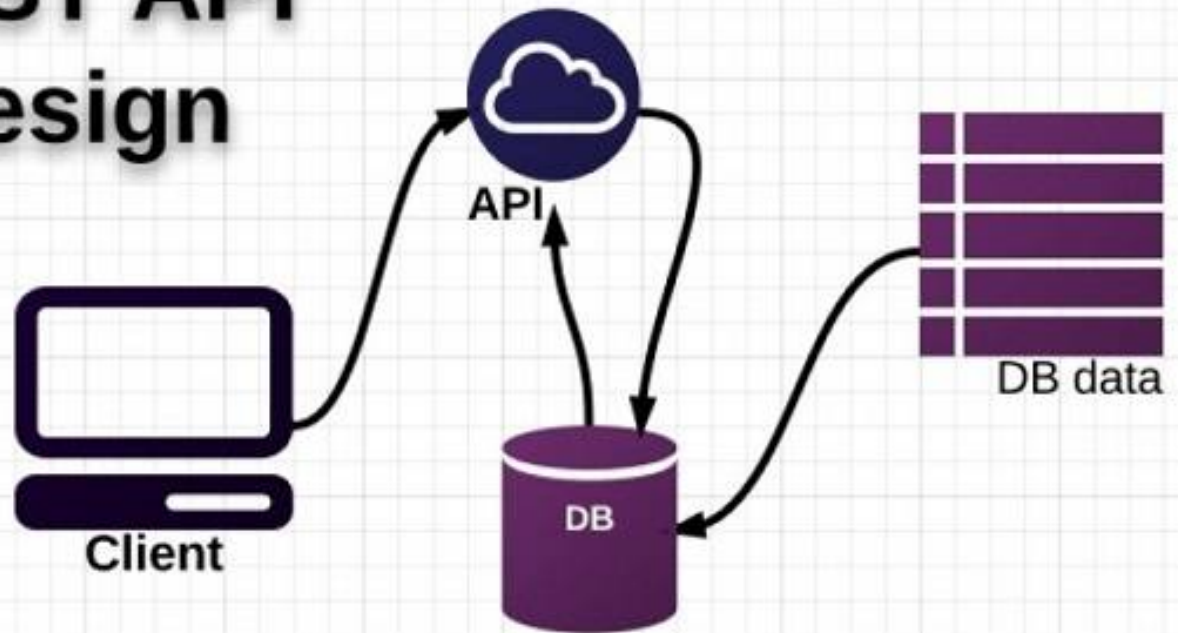
Once upon a time...



Rest API architecture

REST API Design

GET /tasks - display all tasks
POST /tasks - create a new task
GET /tasks/{id} - display a task by ID
PUT /tasks/{id} - update a task by ID
DELETE /tasks/{id} - delete a task by ID



The client can be anything...

Silex micro framework

- Silex is a PHP micro-framework built on Symfony2 components.
- It can be used to build small websites and large applications alike.
- It is concise, extensible, and testable.

[Silex website](#)

How to install

- Create a folder and change its permissions for your user
 - `cd /var/www/html`
 - `sudo mkdir blog`
 - `sudo chown developer:developer blog`
- Download Composer to manage dependencies
 - `cd blog`
 - `sudo curl -sS https://getcomposer.org/installer | sudo php`

How to install

- Create composer.json with Silex dependency
 - nano composer.json

```
{
    "require": {
        "silex/silex": "~1.2"
    }
}
```

How to install

- Download Silex and its dependencies using Composer inside the **blog** folder
 - `php composer.phar update`
- At the end of the update process you should have a **vendor** folder with all the libraries inside

Test the installation

- Create a simple index.php file

```
<?php
require_once __DIR__.'../../vendor/autoload.php';
$app = new Silex\Application();
$app['debug'] = true;
$app->run();
?>
```

- Go to `localhost/html/blog` inside your browser

Rewriting the URLs

- We need to direct every request to our application, index.php
- We need an Apache module, mod_rewrite.
- Edit the configuration file

`/etc/apache2/sites-available/000-default.conf`

`- sudo nano /etc/apache2/sites-available/000-default.conf`

- Insert the following lines

```
<Directory /var/www/html/blog>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```

Create .htaccess

- Enable mod_rewrite
 - sudo a2enmod rewrite
- Create and edit .htaccess file inside blog folder

```
<IfModule mod_rewrite.c>
  Options -MultiViews

  RewriteEngine On
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteRule ^ index.php [QSA,L]
</IfModule>
```

- For an introduction to mod_rewrite

[Mod_rewrite tutorial](#)

Test mod_rewrite

- Restart Apache
 - `sudo systemctl restart apache2.service`
- Go to **localhost/blog** for a successful test, you should see nothing, but it is OK
- Go to **localhost/blog/prova** for a failed test, you should see a long queue of error messages, basically they say that there is no route to /prova

Using an ORM as the database layer

- To keep the example simple, we will use [RedBeanPHP](#), an easy to use ORM (Object Relational Mapping) for PHP.
- The only thing we need is rb.php to include inside our application (no dependencies, only a plain file)
- To use RedBean in our project we need to include rb.php

```
include 'rb.php';
```

RedBean crash course

- To setup the connection

```
R::setup('mysql:host=localhost;dbname=blog',  
        'guest', 'guest');
```

- To create a „bean“

```
$article = R::dispense('article');
```

- To add properties

```
$article->title = 'Brilliance';  
$article->text = 'A good book';
```

- To store a record

```
$id = R::store($article);
```

Exercise

- Edit a file named `create_table.php` and insert the lines of code to create the table 'article', with the following attributes:
 - title
 - text
 - creation_date
- Fill this table with a row for each student of your group (each one has to write his own row)
- Execute the file with the PHP CLI
- Check your work using MySQL Workbench

Retrieving data with Silex

- Come back to index.php
- Include RedBean file
- Add a simple route to a normal web page using the GET method, the first parameter is a pattern, the second parameter is the function called in response (a sort of callback)

```
$app->get('/helloworld',  
function (Silex\Application $app) {  
    //your code here  
})
```

Retrieving data with RedBeanPHP

- To retrieve all records from a table

```
$articles = R::findAll('article');
```

- To retrieve one record from a table

```
$article = R::findOne( 'article',  
    ' id = ? ', [ $id ] );
```

- To transform a bean into an object

```
$obj = $article->export();
```


Responding from Silex

- Reply with a JSON object, it could be a single object or an array of objects or any combination of these categories

```
return new JsonResponse($obj);
```

- Reply with an error message, for example when a resource doesn't exist

```
$app->abort(404, "Article not found");
```

Exercises

- Create a route to a simple HTML page (not a JSON object) with the list of all articles

`/blog/articles.html`

- Create a route to the list of all articles in the form of an array of JSON objects

`/blog/articles`

- Create a route to an article identified by id (see Silex documentation)

`/blog/article/1`