WEB API

THE BASICS

JOÃO RUBENS MARCHETE FILHO

bmarchete@gmail.com

Graduado em Tecnologia em Analise e Desenvolvimento de Sistemas (Unicamp)

Mestrado em Tecnologia e Inovação – Information Visualization (Unicamp)

Licenciado em Informática (Fatec Mogi Mirim)

Licenciado em Matemática (Faculdade Paulista São José)

Pós Graduado em Educação Especial (Faculdade Campos Eliseos)

Licenciado em Pedagogia

Professor – Etec Pedro Ferreira Alves

Professor de tecnologias JAVA/C# – Uniararas

Desenvolvedor de sistemas Web com PHP

Escritor do livro Desenvolvendo um sistema Web com PHP do começo ao fim

AGENDA

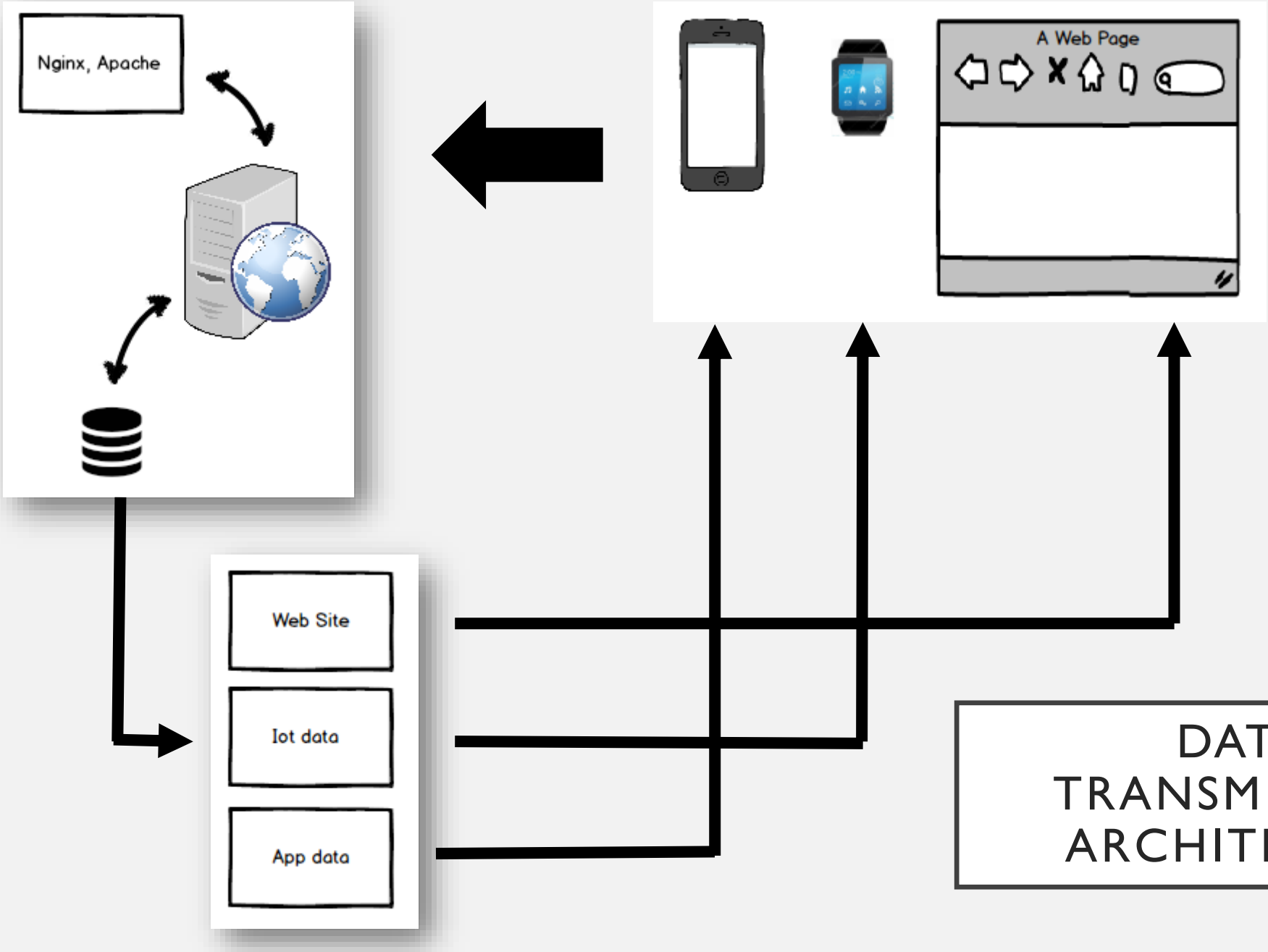
Data Transmission Architecture

Terminologies

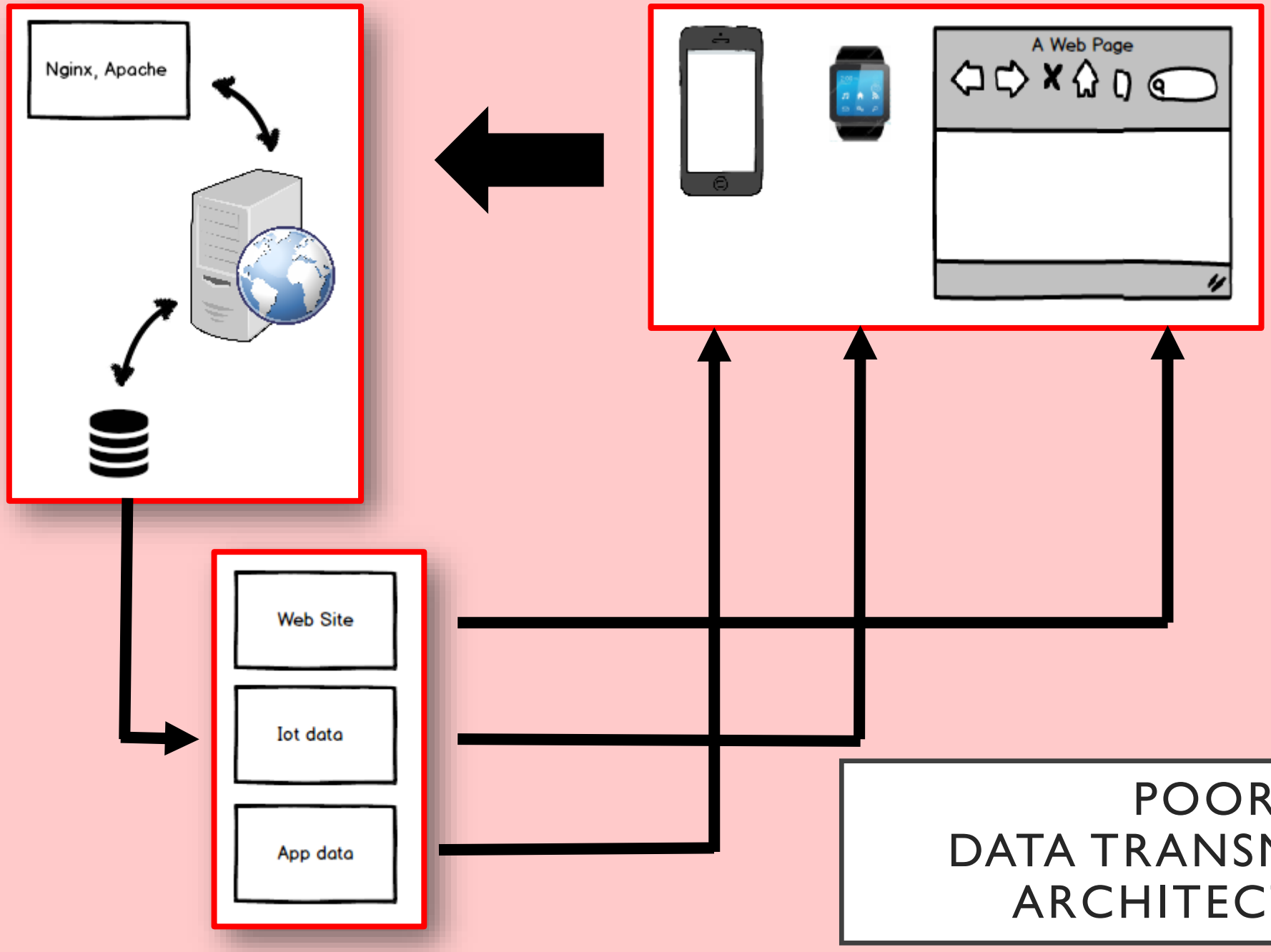
Web Services – SOAP

Web APIs – REST

Making a Web API !



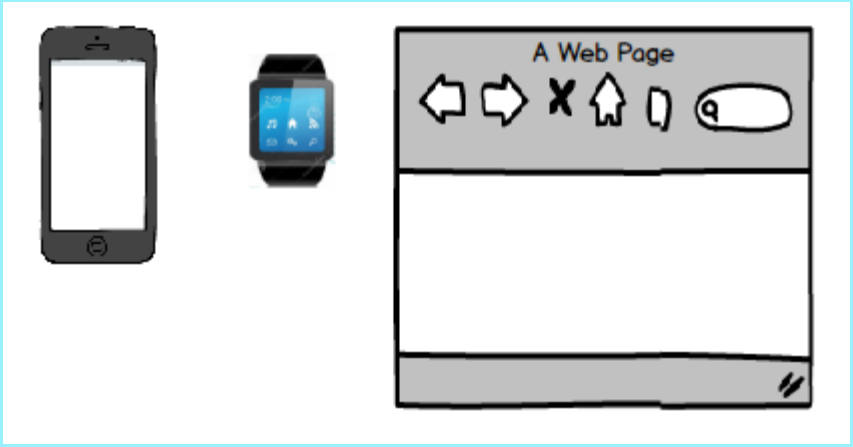
DATA TRANSMISSION ARCHITETURE

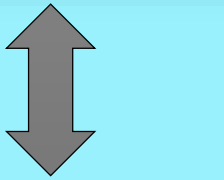


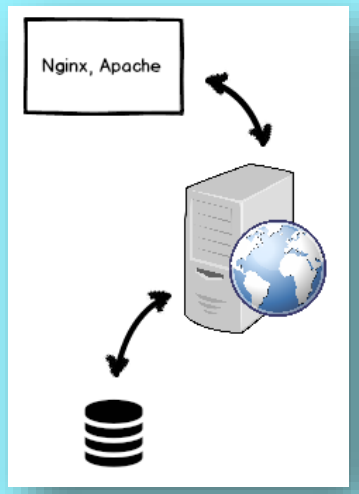
POOR DATA TRANSMISSION ARCHITECTURE

Request

Data







COOL DATA TRANSMISSION ARCHITECTURE

TERMINOLOGIES

- **Web Services** is a open standard based Web applications that interact with other web applications for the purpose of exchanging data

- **Web application** is a client–server software application in which the client (or user interface) runs in a web browser

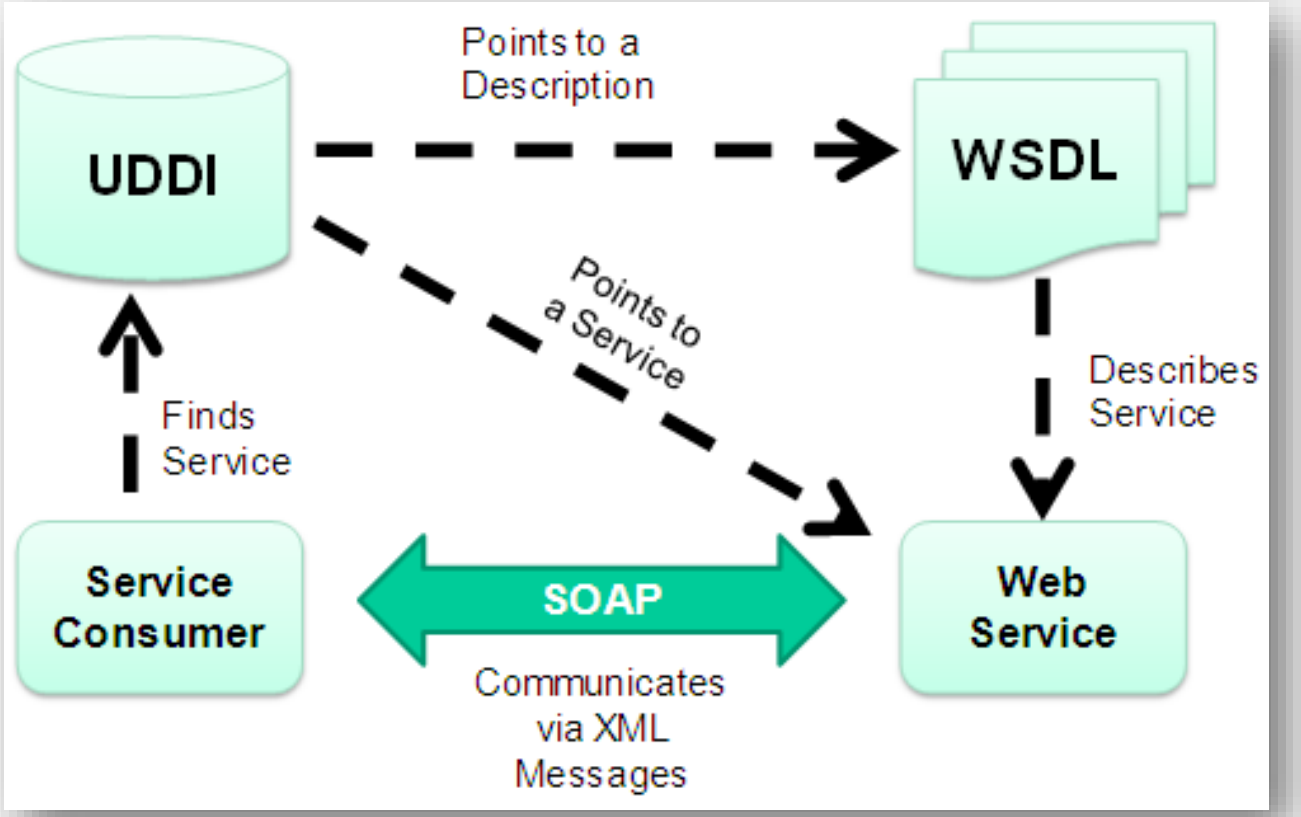
- **Web API** is an application programming interface (API) for either a web server or a web browser

WEB SERVICES

WEB SERVICES - INIT

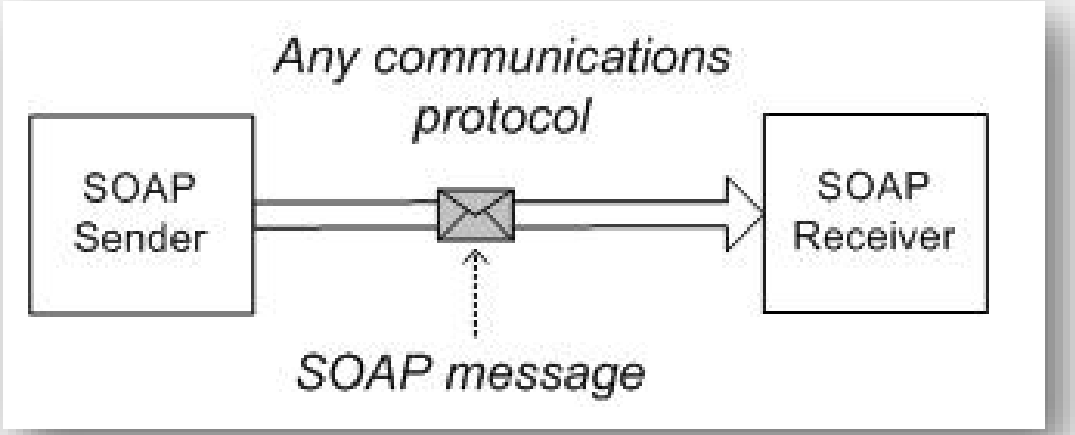
- XML + HTTP (there are others!)
- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

SOAP


Points to a Description
UDDI
WSDL
Points to a Service
Finds Service
Describes Service
Service Consumer
SOAP
Web Service
Communicates via XML Messages

SOAP

- RPC (Remote Procedure Calls)
- Communication Protocol (between applications)
- Designed to communicate via Internet
- Platform independent
- Based on XML
- W3C standard



SIMPLE OBJECT ACCESS PROTOCOL (SOAP)

- “Heavy-Duty" approach
- Focuses on exposing pieces of application logic (not data) as services

- Example:
- switchCategory(User, OldCategory, NewCategory)

REST

REST

- **REST - RE**presentational **S**tate **T**ransfer
- Roy Fielding

- The server simply provides access to resources and the client accesses and presents the resources

WEB SERVICES RELOADED?

- Set of principles that define how Web standards are supposed to be used
- Every component is a resource and a resource is accessed by a common interface using standard methods

- “Light-Weight" approach

REST PRINCIPLES

- Client-Server
  - The clients and the server are separated from each other thus the client is not concerned with the data storage.

- Cacheable
  - Clients can cache the responses

-

REST PRINCIPLES

- Uniform Interface
  - Individual resources are identified using URIs
- Stateless Interactions
  - All of the information necessary to service the request is contained in the URL

HTTP IMPLEMENTS REST

URI - UNIFIED RESOURCE IDENTIFIER

- http://example.com/clients
- http://example.com/clients/243
- http://example.com/clients/create

COMMUNICATION VERBS

- GET
- POST
- PUT
- DELETE
- HEAD
- OPTIONS,
- TRACE

CONTENT TYPES

- XML, HTML, JSON...

```
GET /clients/132 HTTP/1.1
Host: example.com
Accept: text/xml
```

```
GET /clients/132 HTTP/1.1
Host: example.com
Accept: application/x-vcard
```

MAKING A WEB API !

TOOLS

- PHP – Server Side Language
- Embedded server (like Apache or Nginx)
- MySQL shell – create and use databases
- PowerShell – prompt commands
- Postman – API test
- Angular JS Application – client side App
- PhpStorm IDE
- Atom IDE
- Node http-server

LARAVEL - FRAMEWORK

- URIs (REST ?)
- Model View Control (MVC)
- ORM (Eloquent)
- Template Engine (Blade)
- Interface command-line (Artisan)
- Middlewares

SERVER

- MVC Server
- .env configuration

SERVER

- Create a model with database migrations
  - php artisan make:model Client -m
- Migrate the database
  - php artisan migrate

SERVER

- Create the routes (URIs)
  - Route::resource('clients','ClientsController');
- Create the controller to the routes
  - php artisan make:controller ClientController --resource

- Server up

OBRIGADO!

https://github.com/bmarchete/palestra-ft

bmarchete@gmail.com