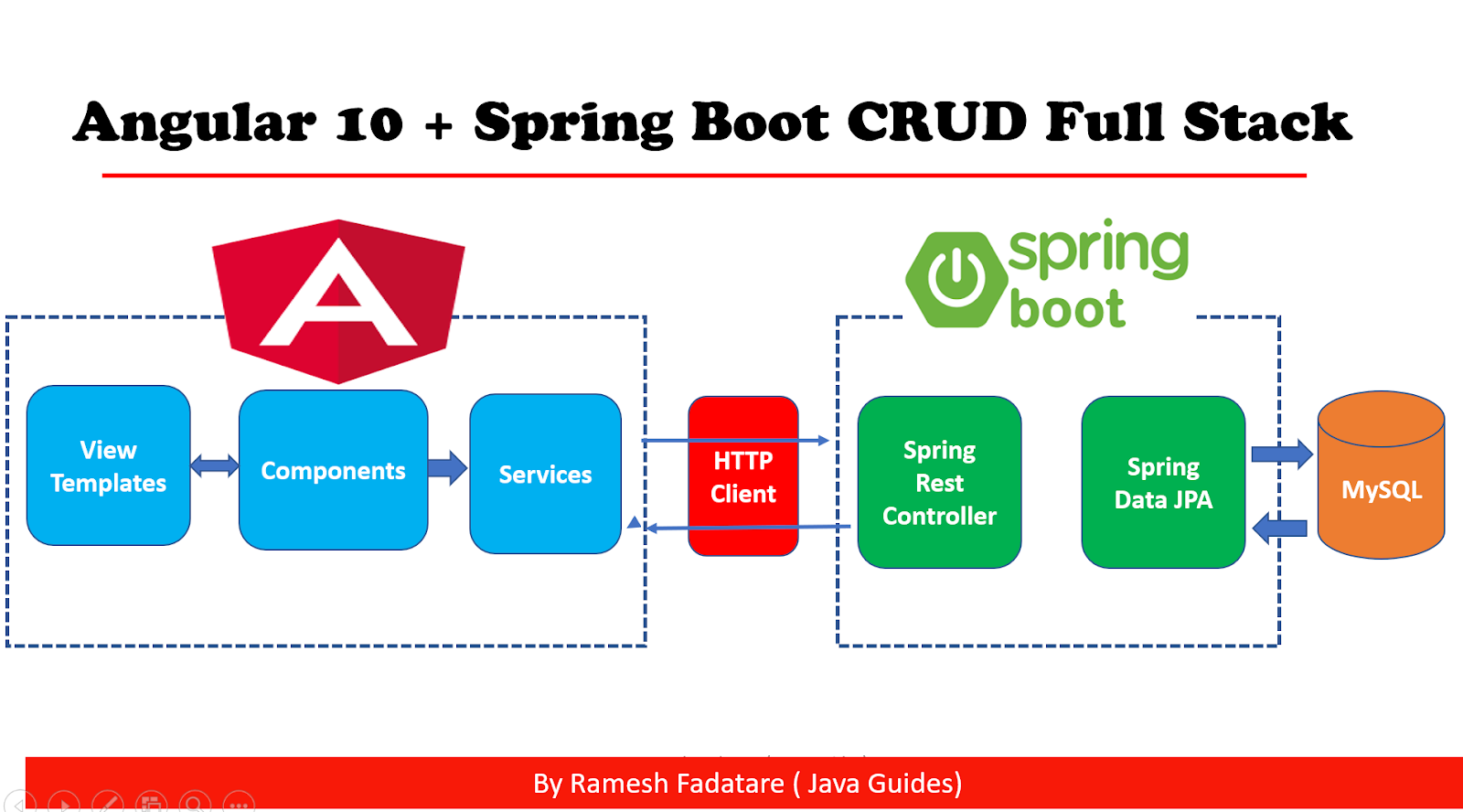
**SERVICES LAYER**

My application will be built by using angular as a front-end technology, spring boost as a server-side framework, and MySQL database as a persistence server. The following endpoints will be exposed by the spring data rest to the HTTP Client of angular. The requested data will be displayed by angular templates on the browser. Pagination support will be provided and only 25 products would be displayed at a time

1. List all products: (URL: <http://localhost:8080/api/products>)

A get request will be sent to the above endpoint to request all the products from the database.

*A GET request*

A picture containing logo

Description automatically generatedA picture containing diagram

Description automatically generated

Get/API/products

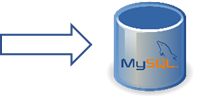
Text

Description automatically generated

1. Product detail (URL: <http://localhost:8080/api/products/:id>)

*A GET request*

When a user clicks an item on the product list page, details of that product are displayed.

Graphical user interface, application

Description automatically generated

Get/API/products/:id

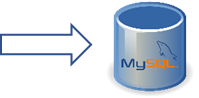
Graphical user interface, text, application

Description automatically generated

1. Search product by category (URL: <http://localhost:8080/api/product-category/:id>)

When a user clicks an item on the product category the respective product details are displayed. For instance, if a user clicks on books, only books will be displayed, or if coffee mugs are selected only related products are displayed.

*A GET request:*

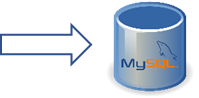


Get/API/findByCategoryId

1. Search product by name: when a user types a string on the search box and hit enter, product will be searched against the database. For instance, when a user searches for a “java book”, products having java in their name property will be displayed.

(URL: <http://localhost:8080/api/products/search/findByNameContaining?name=java>)

*A GET request:*

Shape

Description automatically generated with medium confidence

Get/API/findByName

Text

Description automatically generated

1. **Pagination support:** users will have the option to decide how many products should be displayed in the grid by selecting a number from a drop-down menu. Furthermore, pagination support will be provided to enable users jump to another page and quickly explore products.

*A GET request:*

(URL: <http://localhost:8080/api/products?page=0&size=10>)

A picture containing text, cup

Description automatically generatedA picture containing text

Description automatically generatedA picture containing diagram

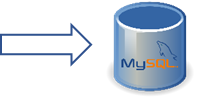
Description automatically generated

Get/API/products?page=0

1. **Save order to the database:** after a user chose a product, decides the quantity, and shipping address, and makes the necessary payments, the order will be stored in the database. A tracking number will be generated randomly and displayed to the user upon successful submission of an order.

(URL: <http://localhost:8080/api/checkout/purchase> )

*A POST request:*

Graphical user interface

Description automatically generated with medium confidence

Post/API/checkout/purchase

changes:

I have added 2 more endpoints to support pagination and save the order to the database