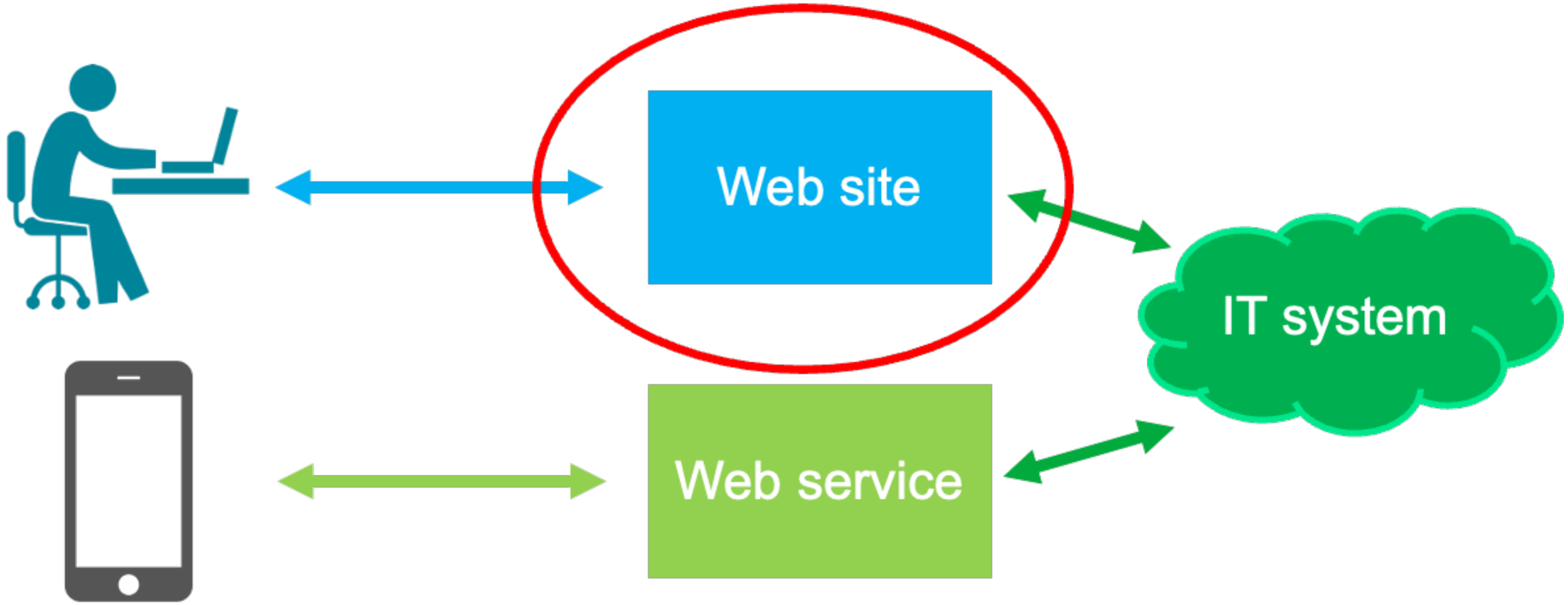
Spring MVC Essentials

Spring MVC - Creating web sites



Spring MVC

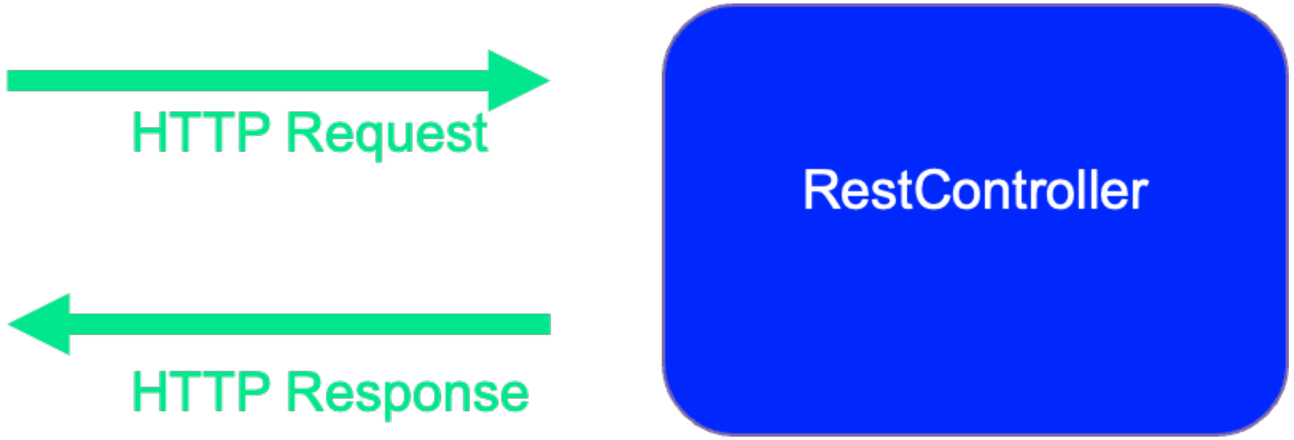
- Spring MVC is a popular web framework for building Java-based web applications
- It is build on top of the core Spring functionality
- Spring MVC follows the Model-View-Controller (MVC) design patter

Spring MVC - Separation of concerns

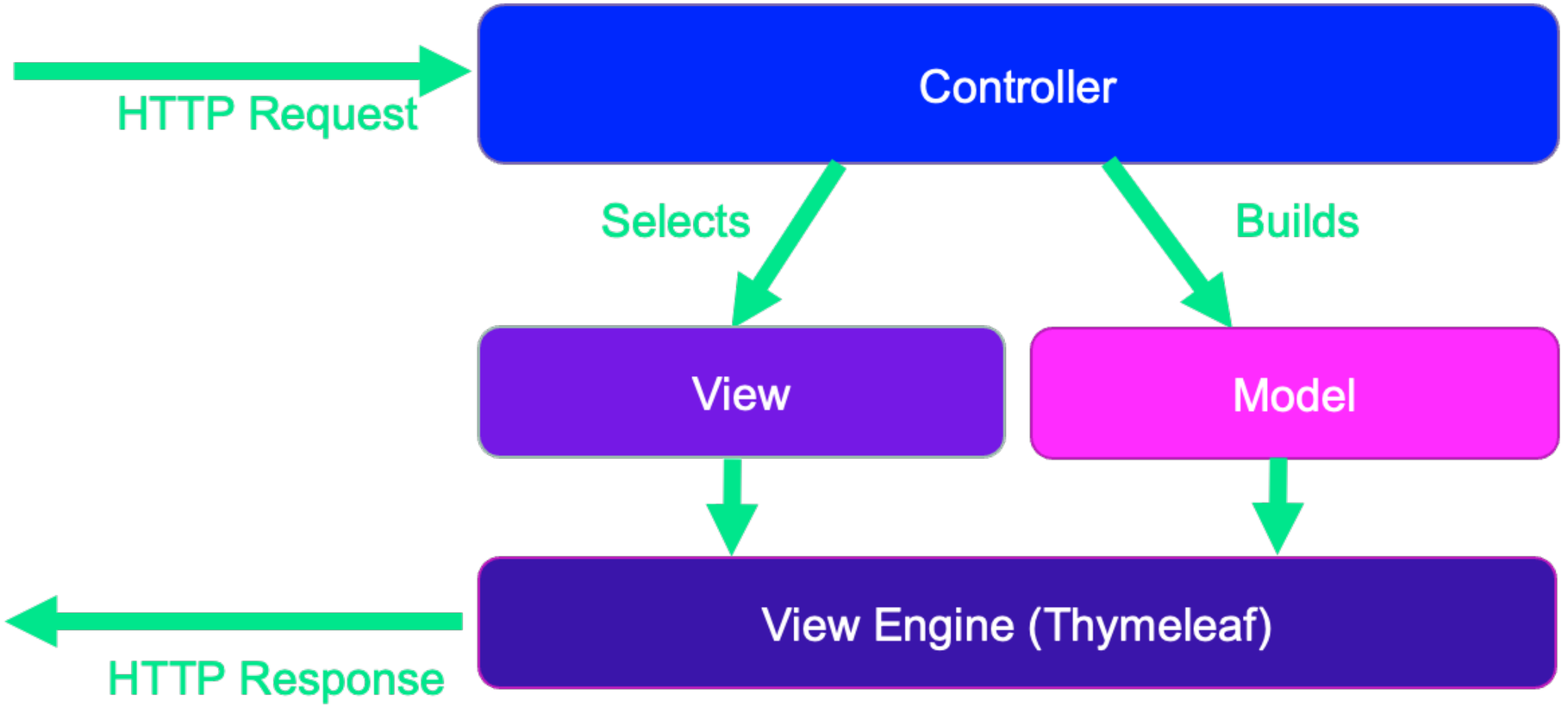
- Spring MVC separates the concerns of an application into three distinct components:
- Model - Represents the data
- View - Represents the user interface
- Controller - Handles user requests, updates the model and selects the view

MVC - Model View Controller

- The RestController from previous exercises is not a MVC Controller
- It acts alone without a Model or a View



MVC - Model View Controller



Spring MVC - The Controller

- Handles user requests, updates the Model and selects the View
- Typically annotated with @Controller which makes it a MVC Controller
- HTTP requests are mapped to methods in the MVC Controller
- These methods often have a Model as an input argument and adds attributes (data) to the Model
- The methods typically returns the name of the selected View

Spring MVC - The Model

- Represents the data, typically in POJOs (Plain Old Java Objects)
- The data is put in the Model by the Controller
- The data is then available to the View
- The Model is the communication vessel between the Controller and the View

Spring MVC - The View

- Represents the user interface
- In Spring MVC typically implemented a templating engine
- Thymeleaf is a popular templating engine used by Spring Boot
- Thymeleaf uses HTML templates to generate HTML pages where attributes from the Model can be mixed into the HTML code

Demo 1 - Creating a Spring MVC app

- Creating the project with the Spring Initializr, selecting the dependencies Web and Thymeleaf
- Creating a MVC Controller
- Creating a Thymeleaf template as the View
- Using a Model to get data from the Controller into the View