CST8288
OOP with Design Patterns
Winter 2024

ALGONQUIN COLLEGE

Week 10 – JSP & MVC

© 2024 - Copyright Algonquin College and other authors as indicated.

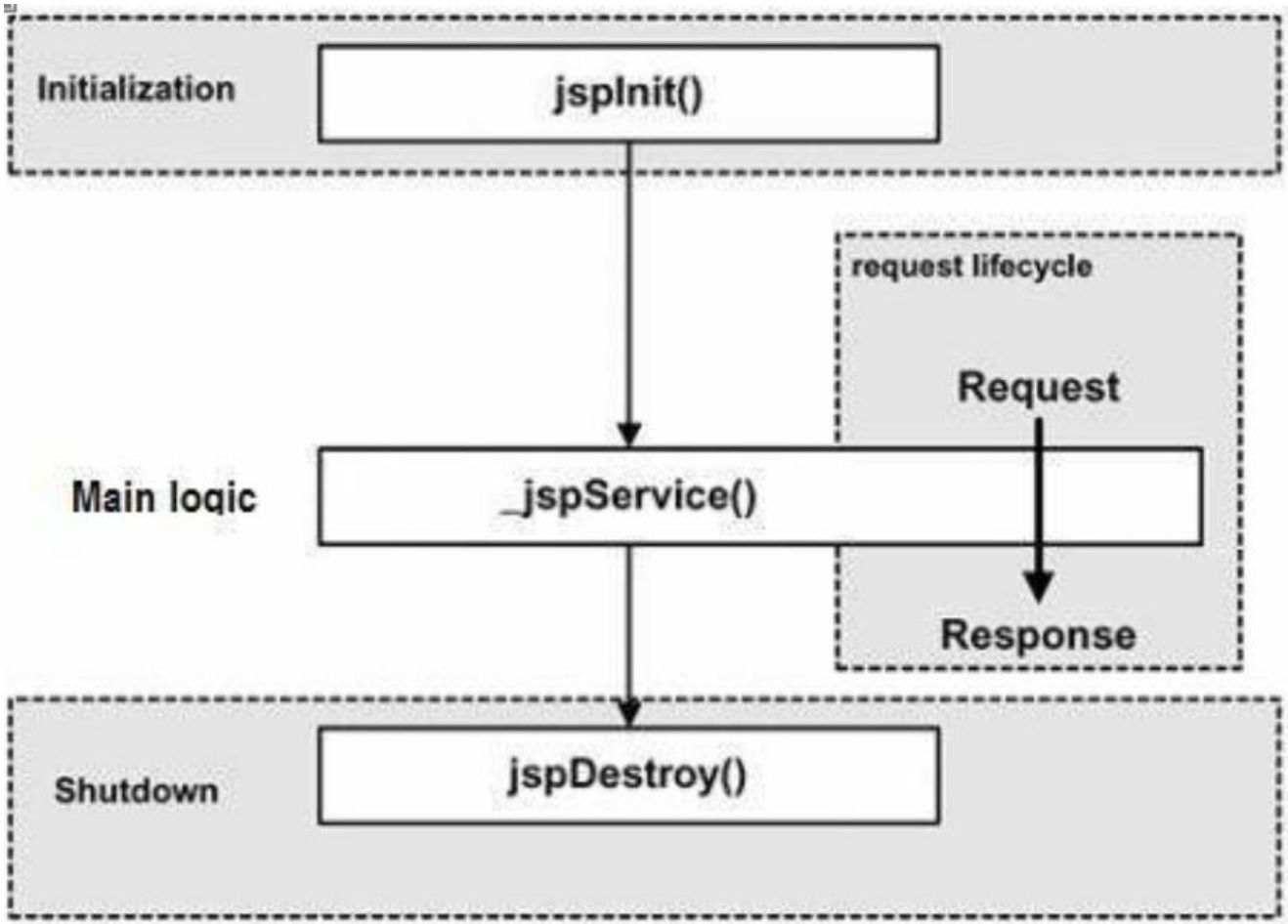
JSP

Java Server Pages

JSP – Java Server Pages

- Java Server Pages (JSP) is a technology for developing Webpages that supports dynamic content.
- Helps developers insert Java code in HTML pages by using special JSP tags. Start with <% and ends with %>.
- Used to create webpage dynamically.
- Built on top of the Java Servlets API. Each JSP page becomes a Servlet when gets deployed.
- Extension .jsp.

JSP Lifecycle



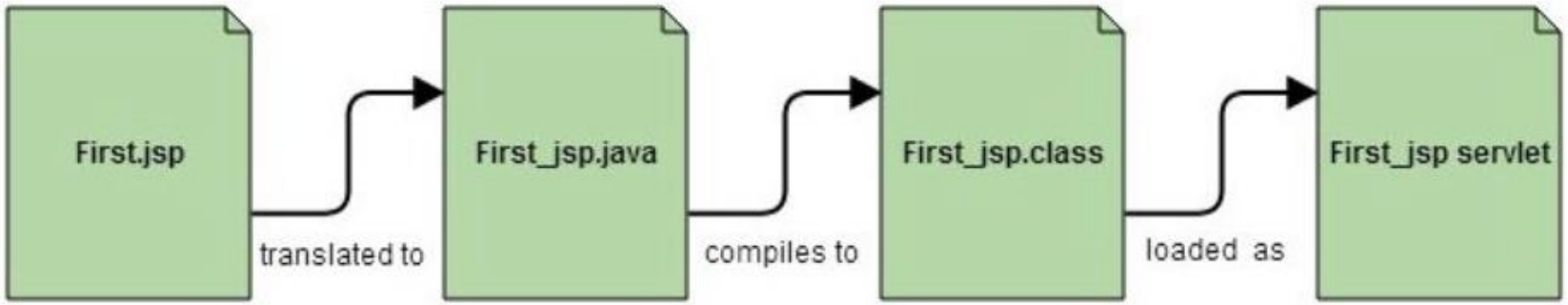
JSP Lifecycle

- Compilation
  - The Servlet engine first checks to see whether it needs to compile the page.
  - Turns the JSP into a Servlet.
  - The Servlet engine compiles the newly created Servlet.
- Initialization
  - Invokes the _jspInit() method before servicing any requests.
  - initialization is performed only once and as with the servlet init method.
- Execution
  - Invokes the _jspService() method in the JSP.
  - The _jspService() method takes an HttpServletRequest and an HttpServletResponse.
- Cleanup
  - JSP is being removed from use by a container.
  - Invokes the _jspDestroy() method which is equivalent to Servlet destroy() method.

THE CONTAINER WRITES THE CORRESPONDING SERVLET FOR YOUR JSP

JSP Under the hood

- The Container takes the JSP pages (.jsp)
- Translate into a Servlet source (.java) file
- Compiles into Servlet class (.class) file
- The run the code exactly same way a typical Servlet runs



JSP Syntax

- Code Fragment - <% code fragment %>
  - <% out.println("Your IP address is " + request.getRemoteAddr()); %>
- Declaration - <%! declaration %>
  - <%! int x = 5; %>
  - <%! Person person = new Person(); %>
- Expression - <%= expression %>
  - Today's date: <%= (new java.util.Date()).toLocaleString()%>
- Comment - <%-- comment --%>
  - <%-- This is a comment in JSP --%>

JSP Directives

- Directives provide directions and instructions to the container, telling it how to handle certain aspects of the JSP processing.
- Directive - *<%@-- directive attribute="value"--%>*
- There are 3 Types of Directives:
- <%@ page attribute="value" %> Defines page-dependent attributes, such as scripting language, error page, and import etc.
  - <%@ page import="com.algonquin.cst8288.* %>
  - <%@ page language="java" %>
- <%@ include attribute="value" %> Includes a file during the translation phase.
  - <%@ include file="header.jsp" %>
  - <%@ include file="footer.jsp" %>
- <%@ taglib attribute="value" %> Declares a tag library, containing custom actions, used in the page
  - <%@ taglib uri="uri" prefix="prefixOfTag" %>
  - <%@ taglib uri="http://www.example.com/custlib" prefix="mytag" %>

MVC
Model-View-Controller

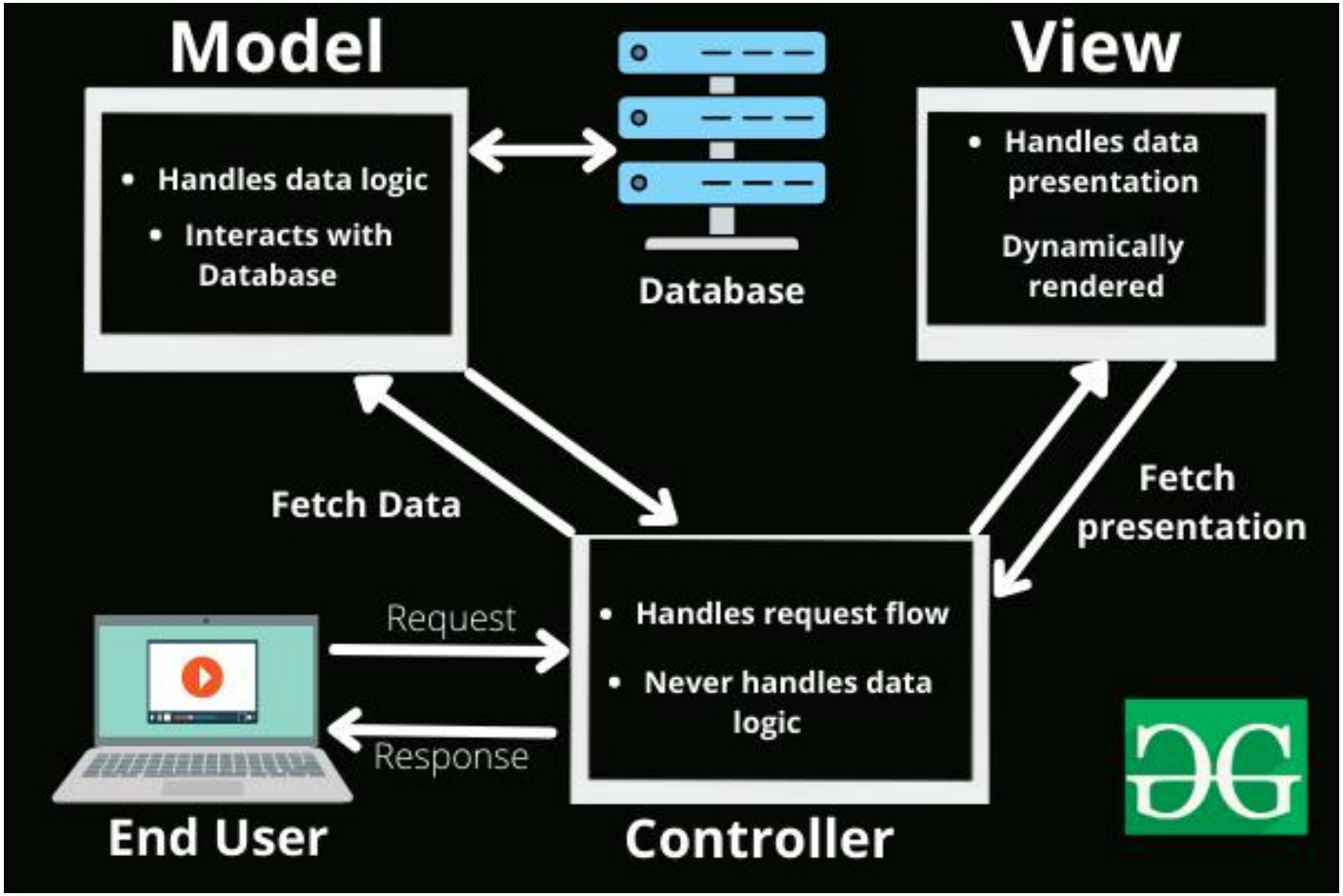
MVC - Model-View-Controller

- MVC is a software architectural pattern commonly used for developing applications with user interfaces.
- It divides the application into three interconnected components: Model, View and Controller.
- Aims to enhance the organization and maintainability of code by separating concerns within an application, making it easier to develop, maintain, and scale software.

Components

- **Model**: Represents the application's data and business logic.
- **View**: Represents the UI and is responsible for presenting the data to the user.
- **Controller**: Orchestrates communication between Model and View. It receives user input, processes it (possibly updating the Model), and instructs the View to update accordingly.

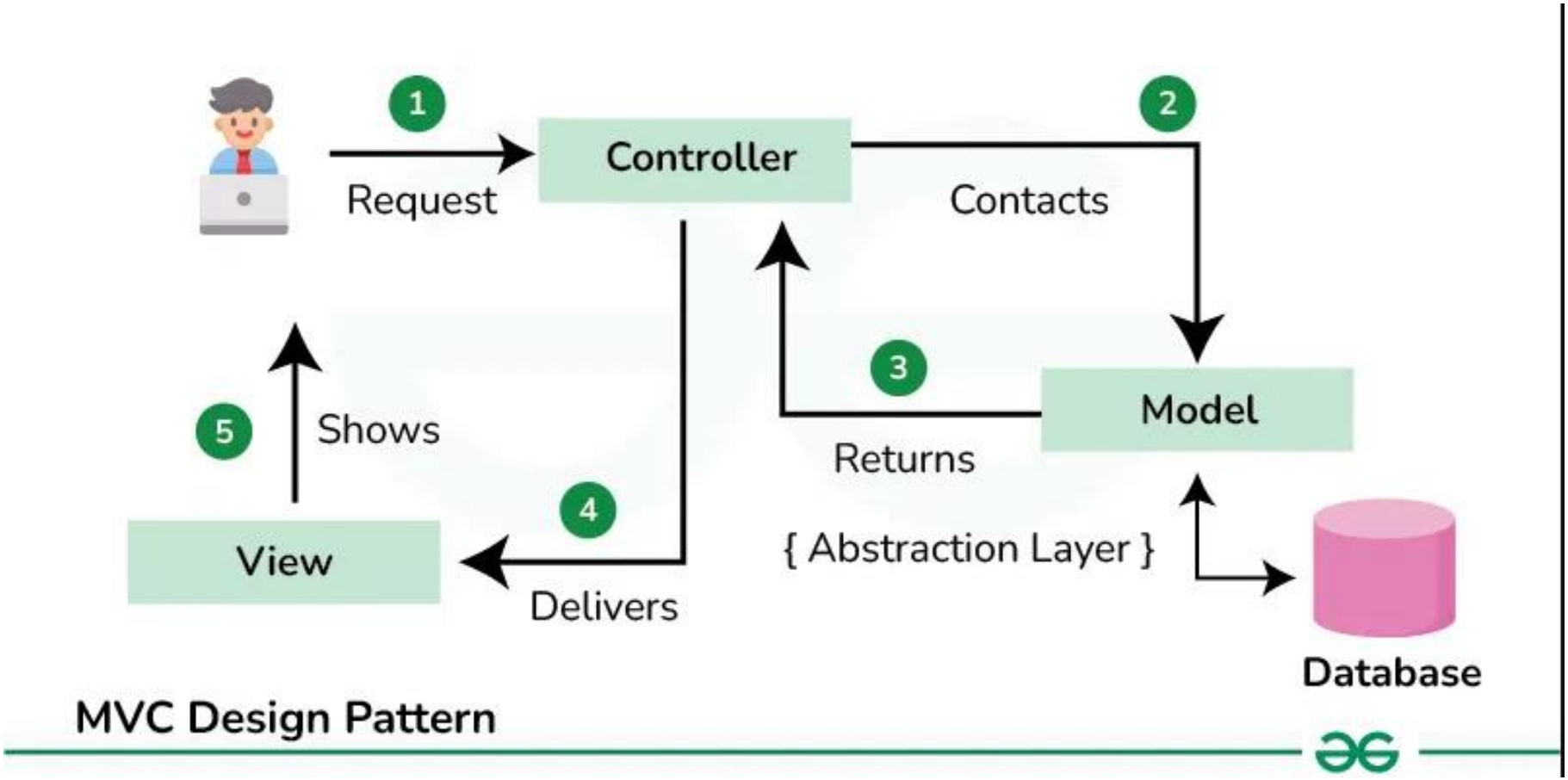
Example Scenario



Code available on Brightspace

by Geeks for Geeks

Example Scenario



Code available on Brightspace

by Geeks for Geeks

Advantages of MVC

- **Separation of Concerns**: Each component has a distinct role, promoting modular and maintainable code.
- **Maintainability**: By dividing the application into distinct components, it's easier to locate, update, or replace specific functionalities without affecting others.
- **Reusability**: Components can be reused in different parts of the application or in other projects.
- **Testability**: Each component can be tested independently, facilitating the testing process.

How MVC Works

1. **User Interaction**:
   - The user interacts with the application, triggering events like button clicks or form submissions.
2. **Controller Receives Input**:
   - The Controller captures user input and decides how the application should respond.
3. **Controller Updates Model**:
   - Based on the input, the Controller updates the Model, which represents the application's data and business logic.
4. **Model Notifies View**:
   - The Model notifies the View of any changes in the data.
5. **View Updates UI**:
   - The View updates the UI to reflect the changes in the data.
6. **User Sees Updated UI**:
   - The user sees the updated UI, completing the cycle.

Thank you!

Credits

- Gustavo Adami
- Sazzad Hossain
- Geeks for Geeks