



# PROGRAMOVÁNÍ WEBOVÝCH APLIKACÍ

MVC, API, REST

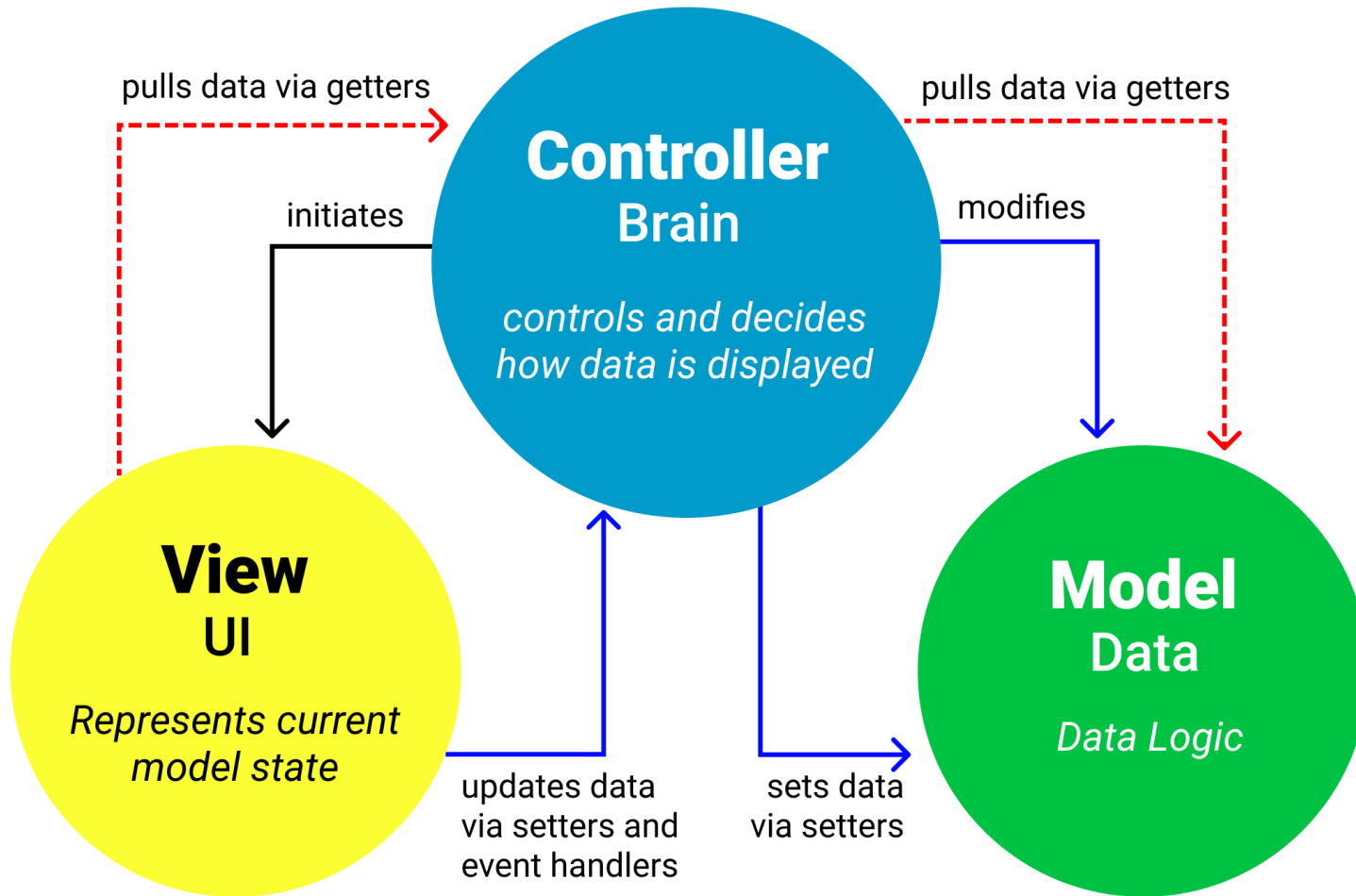
# Application Programming Interface

- Aplikační rozhraní aplikací
- Komunikační prostředek mezi aplikací – serverem, aplikací – uživatelem, ...
- Obsahuje sbírku metod, které rozhraní poskytuje
- Rozhraní neobsahuje samotnou implementaci
- Zobrazuje nám, které metody a procedury lze nad danou třídou/objektem využívat
- Skrze API příslušné metody voláme
- API je obecné pojmenování rozhraní – v závislosti na zdroji pro komunikaci lze API dělit
  - API operačního systému – Windows API, POSIX
  - Grafické API – OpenGL, DirectX
  - Webové API – REST,

# Architektura MVC

- Aplikace je rozdělena do tří samostatných modulů, které spolu jistým způsobem komunikují
- **MODEL** – reprezentace informací, se kterou aplikace pracuje
- **VIEW** – zobrazení uložených dat v pochopitelné podobě pro uživatele
- **CONTROLLER** – řadič, ovládající manipulaci dat způsobených nejčastěji uživatelem
- Typickým příkladem chování aplikace využívající architekturu MVC je libovolná webová aplikace pracující s databází
- Společně s využitím metodik OOP se jedná o často využívanou architekturu ve státním i soukromém sektoru

# MVC Architecture Pattern



21.06.2021

## Architektura MVC

- Model zodpovídá za správu dat
- View uložená data zobrazuje a poskytuje možnost s nimi operovat
- Controller reaguje na události a zařizuje aktualizaci dat pro jejich zobrazení

# Jazyky používané pro jednotlivé části MVC

- **MODEL** – ideálně programovací jazyk, který využívá vlastností OOP
  - Jednotlivé záznamy v databázi jsou reprezentovány jako kolekce objektů
  - Model je reprezentován třídou – atributy třídy odpovídají sloupečkům v tabulce databáze
- **VIEW** – často značkovací jazyk jako je HTML, XML, který nám umožní zobrazit data v přehledné podobě
- **CONTROLLER** - programovací jazyk, který nám umožňuje vytvořit aplikační rozhraní (API)
  - Komunikace s databází – typicky SQL
  - Komunikace s grafickým rozhraním – Kotlin, Python, C#, Java

# Webové API

- Rozhraní webových aplikací pro komunikaci typicky se serverem nebo prohlížečem
- Webové API na serveru obsahuje tzv. **ENDPOINTY**, které na základě jejich volání vrací příslušnou odpověď zpět uživateli
- Odpověď na dané volání je obvykle ve formátu typu **JSON** nebo **XML**
- Odpověď, které uživatel získá jsou pak za pomoci značkovacích jazyků zobrazena v příslušné podobě
- Komunikace probíhá často na základě protokolu HTTP
- **ENDPOINT**
  - Specifikuje, kde se nachází zdroj dat/informací, na které se serveru dotazujeme
  - Přístup ke zdroji je reprezentován pomocí URI, které je skrze HTTP voláno

# REST, CRUD

- **CRUD** je akronymem pro operace **Create**, **Read**, **Update** a **Delete**, které pracují nad databází
  - V SQL se jedná o operace Insert, Select, Update a Delete
- **REST API** – architektura rozhraní, které nám umožňuje vykonávat CRUD operace za pomoci URI
- REST definuje čtyři základní metody volání:
  - **POST, GET, DELETE, PUT**
- Kombinace jedné z metod volání a URI, které odkazuje na zdroj je pomocí HTTP odeslána jako dotaz na server a zpět je serverem zaslána odpověď (JSON, XML)

# Příklady REST API volání

- Mějme situaci, kdy máme webovou aplikaci, které spravuje záznamy knihovny
- Knihovna eviduje jednotlivé knížky, čtenáře a výpůjčky
- REST API, které je navržené nám umožňuje zobrazit všechny položky konkrétního modelu (knížky, čtenáři, výpůjčky) a stejně tak konkrétní položku (knížka, čtenář, výpůjčka)
- **GET** <https://knihovna.cz/api/book> - zobrazení všech uložených knih
- **POST** <https://knihovna.cz/api/reader> - vytvoření nového čtenáře a uložení do databáze
- **PUT** <https://knihovna.cz/api/book/36/> - úprava knížky s ID 36
- **DELETE** <https://knihovna.cz/api/borrow/17/> - odstranění výpůjčky s ID 17



# Kódy odpovědí volání

- Server zasílá odpověď na základě volání, ale neřeší, co je obsahem odpovědi na dotaz
- Společně s daty, které jsou vráceny, jsou v hlavičce těchto dat umístěné kódy reprezentující status odpovědi
- Na základě těchto kódů lze snadno určit, zda bylo volání úspěšné či nikoliv

Kód odpovědi	Status	Popis
<b>200</b>	OK	Volání úspěšně proběhlo
<b>201</b>	Created	Vložení záznamu proběhlo úspěšně
<b>400</b>	Bad Request	Požadavek je neplatný
<b>401</b>	Unauthorized	Nedostatečná oprávnění pro volání
<b>404</b>	Not Found	Volaný zdroj nebyl nalezen
<b>500</b>	Internal Server Error	Požadavek nebyl dokončen pro chybu serveru