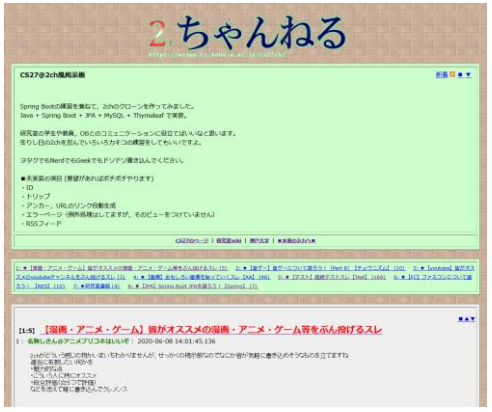
ITスペシャリスト養成コース

第２回
Webアプリケーション開発入門

前回身に付けた知識

- クラウドとは何か
- AWSセミナー
- AWS EC2で
  - apacheでwebページ公開
  - mysql (mariadb) + phpmyadminでdb操作
  - Tomcat+DBを利用したWebアプリ公開



今日の目的

- Webアプリケーションのアーキテクチャを理解
  - Webアプリケーションとは
  - HTTPメソッド
- Spring Bootを利用し簡単なWebアプリを作れる

1日で駆け抜けるしかない Webアプリケーション開発の基礎1

の前に

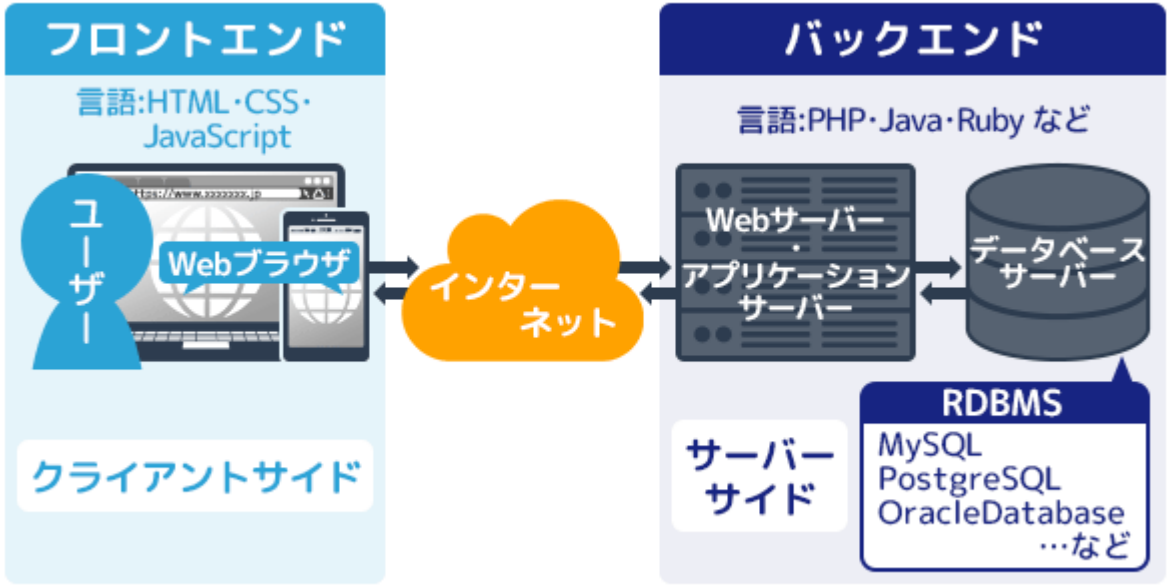
- 今日の作業は二人一組で行います
  - とはいえ，ペアプロはしません
  - 二人で相談しながら進めるためのものです
  - 気分次第でペアが途中で変わります
  - 相互に協力して下さい．足の引っ張りあい禁止
    - だらしねぇという　戒めの心　「ちゃんとやりなさい」
    - 歪みねぇという　賛美の心　「よくやった．すごい」
    - 仕方ないという　許容の心　「まぁしかたねーよ」

ペア　＆　移動

- 強力するけど極力横を向いて直接話さない
  - コロナ対策
- 相手のキーボードには手を出さない
  - 自分でやることに意味がある
  - コロナ対策
- ペアで解決できなくても残り２ペアいます
  - 寝たら起そう

Webアプリケーション

- Webサーバ上で動作するアプリケーション
- Webブラウザを用いて利用
- Web経由でクライアントがサーバにアクセスする



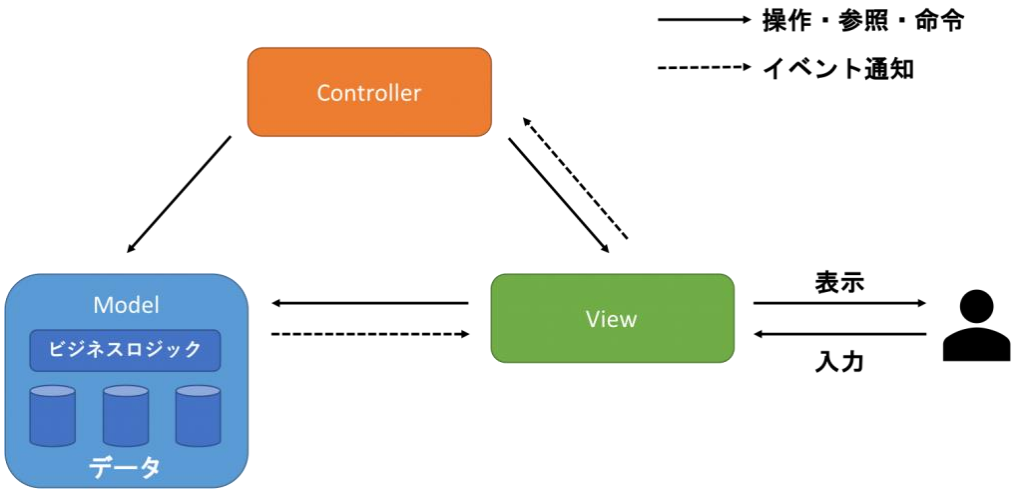
一般的なWebアプリケーションの構成

似たような言葉

- Webサービス
  - 広義：Web上で提供されるサービス
  - 狭義： HTTPやXMLなどWebコンテンツの送受信に用いられる標準技術を基盤に、ソフトウェア間でデータや機能を連携できるようにしたもの
- WebAPI
  - （狭義の）Webサービスを利用するためのインタフェース．ほぼWebサービスと同義．
- Webシステム
  - Webを利用したもの．ここまでのWebxxx全部．

MVC（オリジナル）

- Smalltalkで生まれた概念
- アプリケーションを，Model, View, Controllerに分けるソフトウェアのデザインパターン（概念）

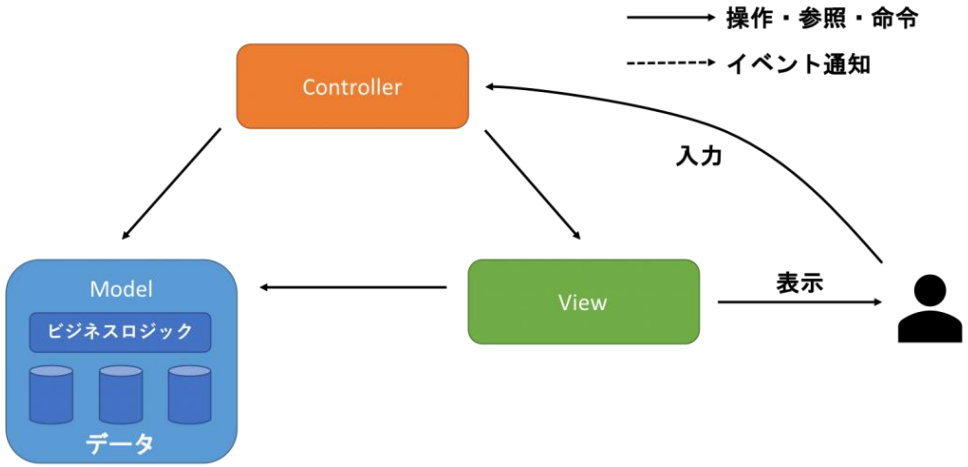


Model: データとビジネスロジックを担当．データに変更があったら，ビューに通知
View: Modelからデータを取り出し，整形して出力．ユーザーから入力があったら，コントローラに通知
Controller: Viewからの入力を受けて，Modelに入力を伝達．また，ビューに表示命令を出す

https://eejanaica.com/need-attention-mvc-of-web-application/

MVC （Webアプリ）

- データに変更があったら，Viewに通知する
- ユーザーから入力があったら，コントローラに通知　　→ Webアプリでは難しい



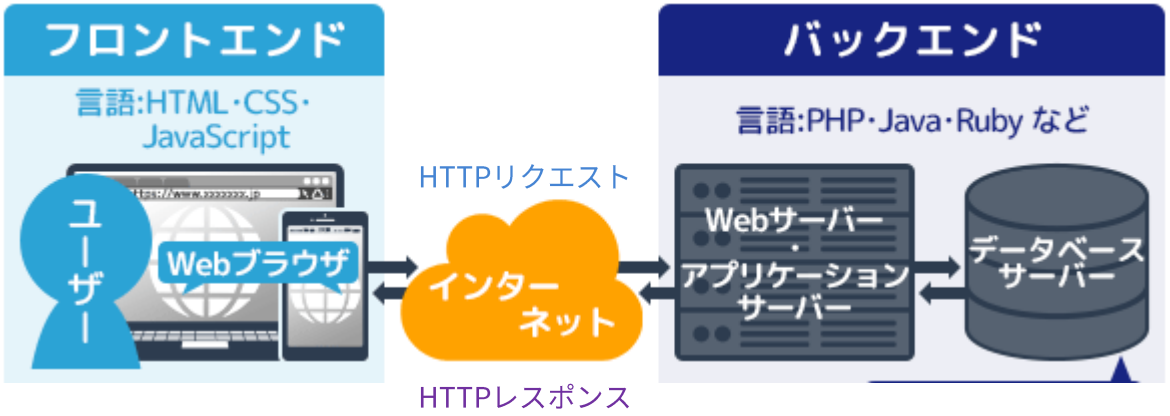
Model:　　データとビジネスロジックを担当．
View:　　Modelからデータを取り出し，整形して出力．
Controller:　ユーザーからの入力を受けて，Modelに入力を伝達．また，Viewに表示命令を出す．

ここからのMVCは全てWebアプリの世界の話

https://eejanaica.com/need-attention-mvc-of-web-application/

HTTPアクセス

- Webアプリケーションへのアクセスは基本的にブラウザから行う　→　HTTPによるアクセス



Webサーバに対し，HTTPリクエストメソッドで機能の要求
リクエストに対し，HTTPレスポンスをサーバは返答

リソース指向アーキテクチャ（ROA）

- それ自体を参照するに値するものをリソースとして定義し，リソースを中心に考えるアーキテクチャ
- ROAに則り実装されたWebサービス：RESTfull API
- ROAの概念
  - 名前：リソースは少なくとも1つのURIを持っていなければならない
  - 表現：Webサーバはリソースを特定の言語(英語，日本語など)，フォーマット(XML，JSONなど)で送信しなければならない
  - リソース間のリンク：リソースには別のリソースへのリンクを含めることができる．リンクを含めることで別のリソースに接続することができる．

詳しくは　https://qiita.com/NagaokaKenichi/items/0f3a55e422d5cc9f1b9c

HTTPリクエストメソッド

- Webのリソースに対する要求　全８種類

メソッド	意味
GET	リソースの取得
POST	子リソースの作成 リソースへのデータ追加 その他処理
PUT	リソースの更新 リソースの作成
DELETE	リソースの削除
HEAD	リソースのヘッダ (メタデータの取得)
OPTIONS	リソースがサポートしているメソッドの取得
TRACE	プロキシ動作の確認
CONNECT	プロキシ動作のトンネル接続への変更

HTTPリクエストメソッド

- Webのリソースに対する要求　全８種類

メソッド	意味
GET	リソースの取得
POST	子リソースの作成 リソースへのデータ追加 その他処理
PUT	リソースの更新 リソースの作成
DELETE	リソースの削除
HEAD	リソースのヘッダ (メタデータの取得)
OPTIONS	リソースがサポートしているメソッドの取得
TRACE	プロキシ動作の確認
CONNECT	プロキシ動作のトンネル接続への変更

リソースに対するCRUDに対応

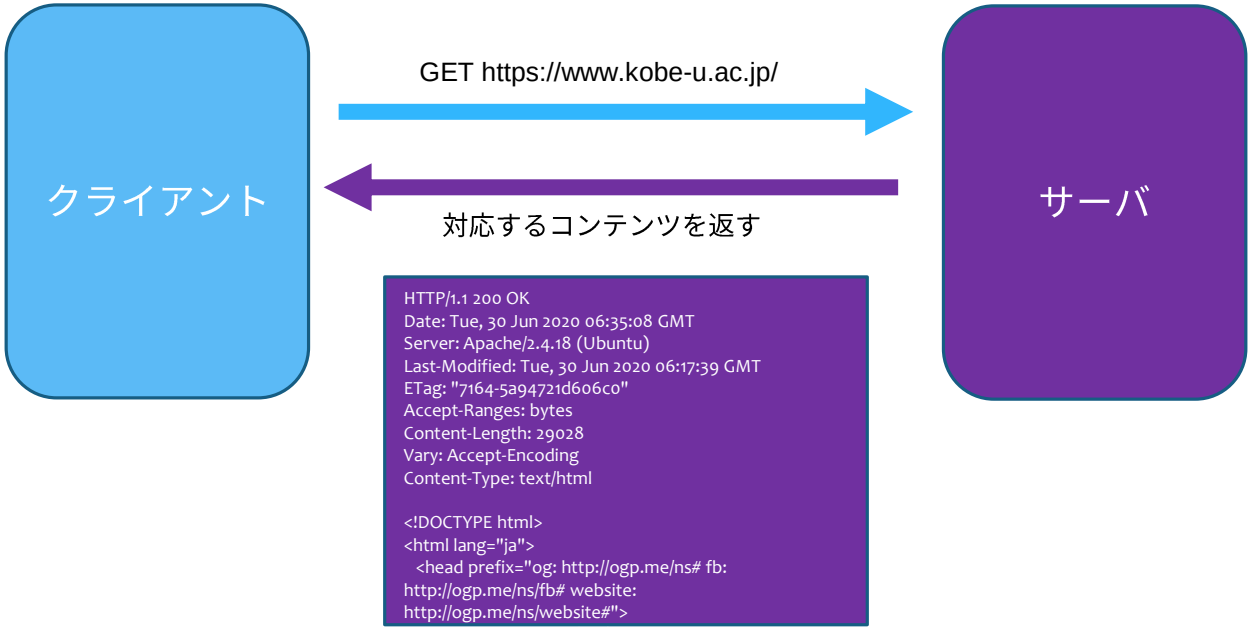
HTTPメソッド（CRUD）

CRUD名	意味	メソッド
Create	作成	POST/PUT
Read	読み込み	GET
Update	更新	PUT
Delete	削除	DELETE

Webアプリケーション実装に必要となるデータ操作を実現

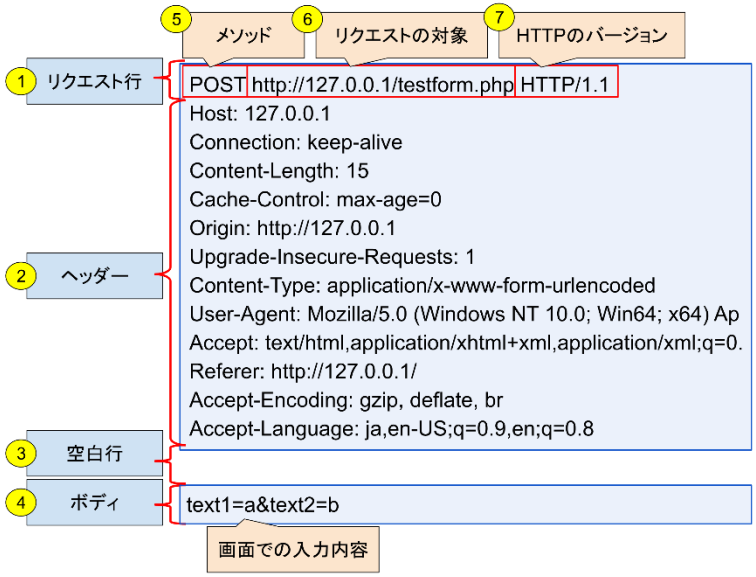
GET：リソースの取得

- 指定したURIの情報を取得するメソッド
  - ブラウザでURIを入れてアクセスする動作
  - 呼び出しに成功すると，対応するリソースを返却



POST：リソースの作成，追加

- 子リソースの作成
- リソースへのデータの追加
- 他のメソッドでは対応できない処理
  - 本来GETで出来るが，文字数制限で・・・等
- HTTPリクエストのボディで内容を送る



https://itsakura.com/network-http-get-post

PUT：リソースの更新，作成

基本的には利用しない
RESTで作成する場合に，利用しないこともない

DELETE：リソースの削除

基本的には利用しない
RESTで作成する場合に，利用しないこともない

HTTPレスポンス

- HTTPリクエストに対するサーバのレスポンス
  - 番号によってそれぞれ意味が異なる
    - 1xx：情報
    - 2xx：成功
    - 3xx：リダイレクション
    - 4xx：クライアントエラー
    - 5xx：サーバエラー
  - 詳しくは

https://developer.mozilla.org/ja/docs/Web/HTTP/Status

URLパラメータ

- HTTPメソッドを発行する際のパラメータ
  - サーバに特定の情報を送りたい
    - ID, 名前, 文字列, ・・・

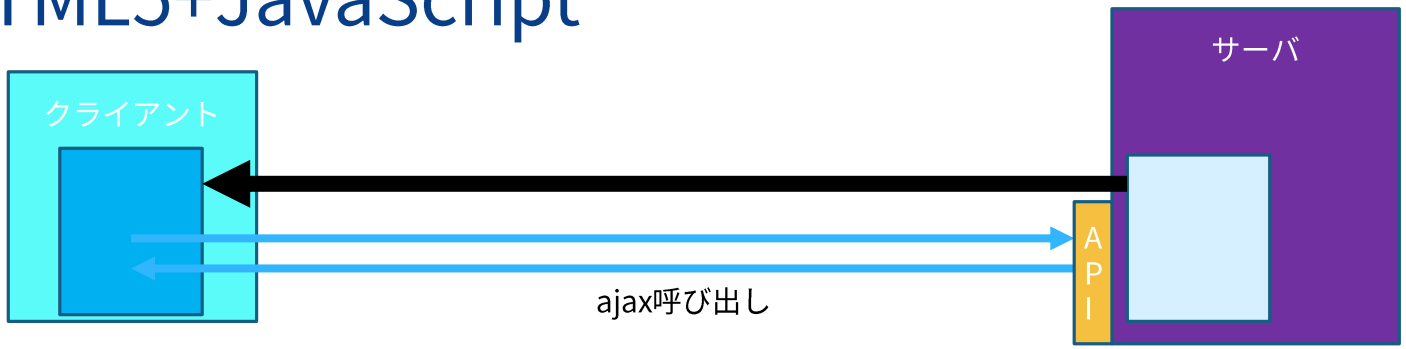
- Request Parameter
  - URLの後ろにクエリとして付け加える
    - server/person?name=hoge&age=20

- Path Parameter
  - URLそのものがパラメータになる
    - server/person/hoge/name

Webアプリ実装のパターン

- サーバ側でViewを生成するパターン
  - CGI, JSP,など昔ながらの方法

- クライアント側でViewを生成するパターン
  - HTML5+JavaScript



Webアプリケーションの本質

- 目的とするWeb上のリソースに対し，様々なHTTPリクエストを用い操作を行えるインタフェースを提供する
- クライアントからの要求に対し，適切なレスポンスコード，コンテンツを返す

HTTPリクエストを送ってみる

- 手元にcurlがあればそれで
- 無い場合は前回のec2インスタンスにログイン

- 基本操作：curl [option] [URL]
  - i レスポンスヘッダを出力
  - v リクエスト・レスポンス　ヘッダを出力
  - X POSTリクエスト
    - curl -X POST [url] -d "name=hoge"

POSTMAN

はじめてのSpring Boot

Webアプリ開発

- 業務（Enterprise）アプリでどのような言語が使われているか

Now We're Talking

Over the last nine months, Java and JavaScript[2] have remained the two dominant languages in the enterprise developer landscape throughout two rounds of research. Industry research firm RedMonk published language rankings in March 2018 that also placed Java and JavaScript at the top tier of development languages–corroborating our findings that Java is alive and well.

In contrast to our findings, however, RedMonk found Python and PHP used more frequently than C# and C++, but only marginally. Indeed, RedMonk's Stephen O'Grady writes that "the numerical ranking is substantially less relevant than the language's tier or grouping."

Overall, IT Decision Makers (ITDMs) report using over 25 total languages–but over half of those languages are used so infrequently as to receive a single digit percentage. In November 2017, ITDMs reported their language preferences, the top six of which are used a significant portion of the time, while many other languages being used for application development are employed at lower rates.

The ordinal ranking and clear separation between Java and JavaScript and all other languages was even more stark in research from March 2018.

LANGUAGE	NOV 2017	MAR 2018	NOV-MAR
Java	57	58	+1
JavaScript	54	57	+3
C++	45	46	+1
C#	28	26	-2
Python	30	25	-5
PHP	22	22	--
VB.NET	19	17	-2
C	19	16	-3
Visual Basic 6	18	16	-2
VBA	16	15	-1
Perl	16	12	-4
Ruby	9	7	-2
Swift	6	6	--
Typescript	5	5	--
Objective-C	6	5	-1
Assembly	6	4	-2
Matlab	7	4	-3
R	4	3	-1
Scala	5	3	-2
Go	3	2	-1
Groovy	3	2	-1
Haskell	2	2	--
CoffeeScript	2	1	-1
Lua	3	1	-2
Other	3	3	--

2 JavaScript is needed for many web applications. Question did not differentiate between server-side JavaScript (i.e.: Node.js) and browser-based use of the language.

1. Java
2. Java Script
3. C++
4. C#
5. Python

Cloud Foundry Foundation: https://www.cloudfoundry.org/wp-content/uploads/Developer-Language-Report_FINAL.pdf

JavaによるWebアプリ開発

- J2EE: Sun Microsystemsが策定したJavaによる企業システム構築のための仕様
- Java業務アプリケーションのデファクト
- Java1.5以降はJavaEEと改名 → JakartaEE

- 中心機能
  - EJB: ビジネスロジックを実装するオブジェクト
  - Servlet: Webサーバ上で動作するJavaプログラムの仕様を定めた標準の一つ
  - JSP: Webページ内にJavaプログラムを埋め込み、これをサーバ上で実行して結果を反映したページを動的に生成することができる技術

EJBの仕様が肥大化

■設定方法が複雑，定義が大変，DBパフォーマンス

Rod Johnson
オーストラリアの天才プログラマ

JavaWorld DAY 2005, TOKYO

エンティティBean（EJB：含まれるデータベース・アクセスのカプセル化機能）
なんてないほうがよかった。エンティティBeanのせいで2〜3年が無駄に失われてしまった

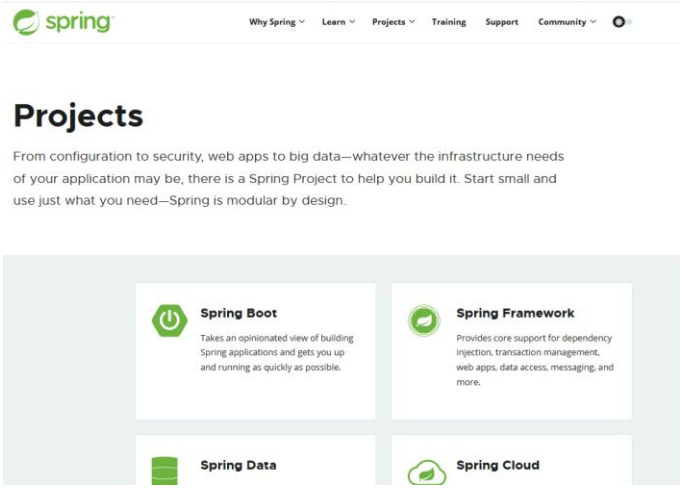
新規のプロジェクトがエンティティBeanを採用したという話は，もはや1件も聞かない

TopLinkはうまく動いていたが，
エンティティBeanは3年もかけてうまく動かないことがわかっただけ

TopLinkはベンダーによる囲い込みという弊害はあるが，
それでも動かないEJBよりはマシ．悪い標準は，標準がないことよりも悪い

Springプロジェクト

- 2004 Rod Johnsonによりリリース
  - Java向けのOSSフレームワーク
    - 24種類の Spring プロジェクトが存在（2020）
  - Spring FrameworkがCore
    - 当初はDI実現のための小さなフレームワーク



https://spring.io/projects

Spring Boot

- Springのプロジェクト
  - Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can “just run”
  - 巨大化して分かりにくいSpring Frameworkをベースに手軽にWebアプリを開発できるフレームワーク
    - Spring MVCを用いたWebアプリ作成

Hello Spring Boot

SpringBootの仕組みの基本1

- 前提知識
  - SpringBoot/各レイヤの責務
  - SpringBoot/ DTO

環境確認

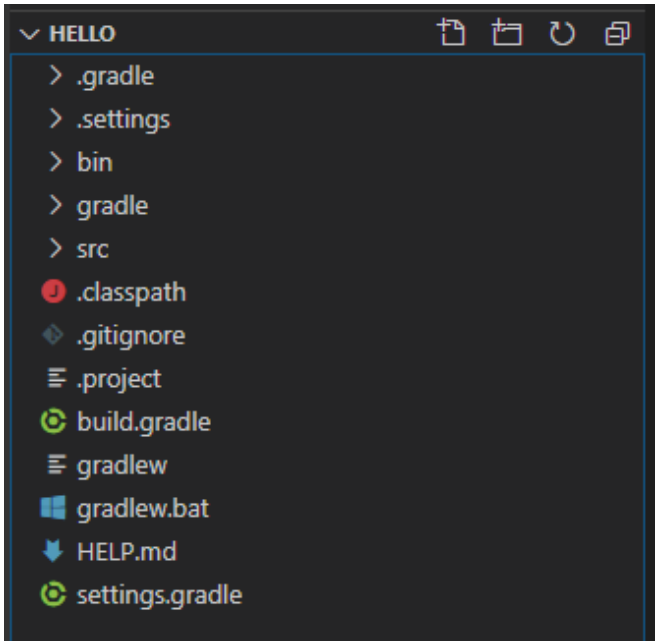
- Visual Studio Code with Java Env
  - 大規模ソフトウェア論で導入済み？
    - JDK14

- Spring Boot用 VSCode extension
  - Spring Boot Tools
  - Spring Boot Dashboard
  - Spring Boot Extension Pack

Spring Initializer

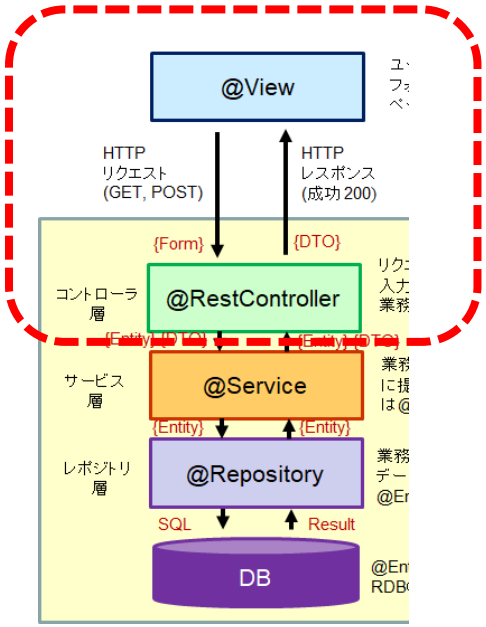
- Spring Bootのプロジェクト雛形を自動生成
  - https://start.spring.io/

- ダウンロードしたzipを展開するとそのままプロジェクトになる



実装

- 講義wiki 第2回/HelloSpringBoot を参照
- Controller/Rest Controllerで文字列表示（だけ）
  - それぞれ Request Param / Path Param を実装
- 簡単な動作確認



実装

Helloプロジェクト

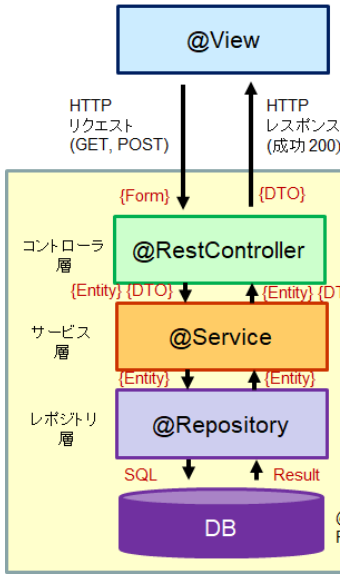
- まとめ
  - SpringBootプロジェクトを０から構成できる
  - ビルド，実行が出来る
  - Request/Path Parameterを理解
  - Controller/RestControllerの違いを理解

一通り簡易Webアプリを作る

- ToDo管理システム

ToDo管理アプリ

- 講義wiki　第2回/ToDo管理　を参照
- View⇔Controller ⇔ Service ⇔ Repositoryを経由する一通りのWebアプリ
- テンプレート型Webアプリ
- Viewからデータを POSTで受け取る
- ViewでRepositoryからのデータを表示する



SpringBootの仕組みの基本2

- 前提知識
  - SpringBoot/Thymeleaf
  - SpringBoot/JPA
  - SpringBoot/関連技術情報

  - SpringBoot/バリデーション
  - SpringBoot/例外処理

APIのテスト

- POSTメソッドのテストはブラウザからでは困難
- 毎回curlなどでやるのも大変
- GUIツールを使いましょう
  - POSTMAN：https://www.postman.com/

実装