

# 掲示板システム作成課題

ex-bbsという名前の

- Spring starter project
  - リモートリポジトリ(今回もGitを使用します)
- を作成してから開始してください。

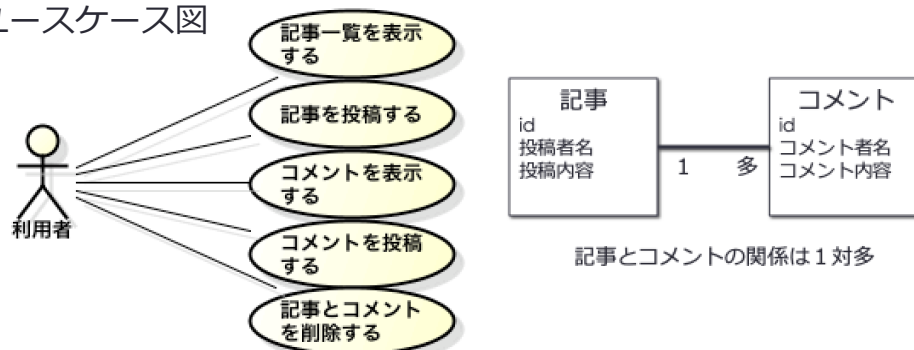
Ver.202010

作成者：伊賀 将之

## 作成するもの

- 1つの投稿内容に複数のコメントをつけられる掲示板
- 記事とコメントは新しい投稿が上に来るように作成する
- 記事削除ボタンが押されたら記事に関連付いているコメントごと削除する

ユースケース図



投稿者名:	<input type="text"/>
投稿内容:	<div><div></div></div>
<input type="button" value="記事投稿"/>	
投稿者名: 山田	
投稿内容:	
新たな投稿です。	
<input type="button" value="記事削除"/>	
コメント者名: 吉田1	
コメント内容:	
山田さん書き込みのコメント1	
コメント者名: 吉田2	
コメント内容:	
山田さん書き込みのコメント2	
名前:	<input type="text"/>
コメント:	<div><div></div></div>
<input type="button" value="コメント投稿"/>	
投稿者名: 伊賀	
投稿内容:	
この掲示板について	
<input type="button" value="記事削除"/>	
コメント者名: 佐藤1	

## DDL 別途配布します

```
CREATE TABLE articles
(
  id serial NOT NULL, --投稿ID
  name text NOT NULL, --名前
  content text NOT NULL, --コンテンツ
  CONSTRAINT articles_pkey PRIMARY KEY (id)
);
CREATE TABLE comments
(
  id serial NOT NULL, --コメントID
  name text NOT NULL, --名前
  content text NOT NULL, --コンテンツ
  article_id integer NOT NULL, --投稿ID
  CONSTRAINT comments_pkey PRIMARY KEY (id),
  FOREIGN KEY (article_id) REFERENCES articles (id)
);
```

## DML 別途配布します

```
INSERT INTO articles(name, content) VALUES('伊賀', 'この掲示板について');
INSERT INTO articles(name, content) VALUES('山田', '新たな投稿です。');
INSERT INTO comments(name, content, article_id) VALUES('佐藤1', '伊賀さん書き込みのコメント1', 1);
INSERT INTO comments(name, content, article_id) VALUES('佐藤2', '伊賀さん書き込みのコメント2', 1);
INSERT INTO comments(name, content, article_id) VALUES('佐藤3', '伊賀さん書き込みのコメント3', 1);
INSERT INTO comments(name, content, article_id) VALUES('吉田1', '山田さん書き込みのコメント1', 2);
INSERT INTO comments(name, content, article_id) VALUES('吉田2', '山田さん書き込みのコメント2', 2);
```

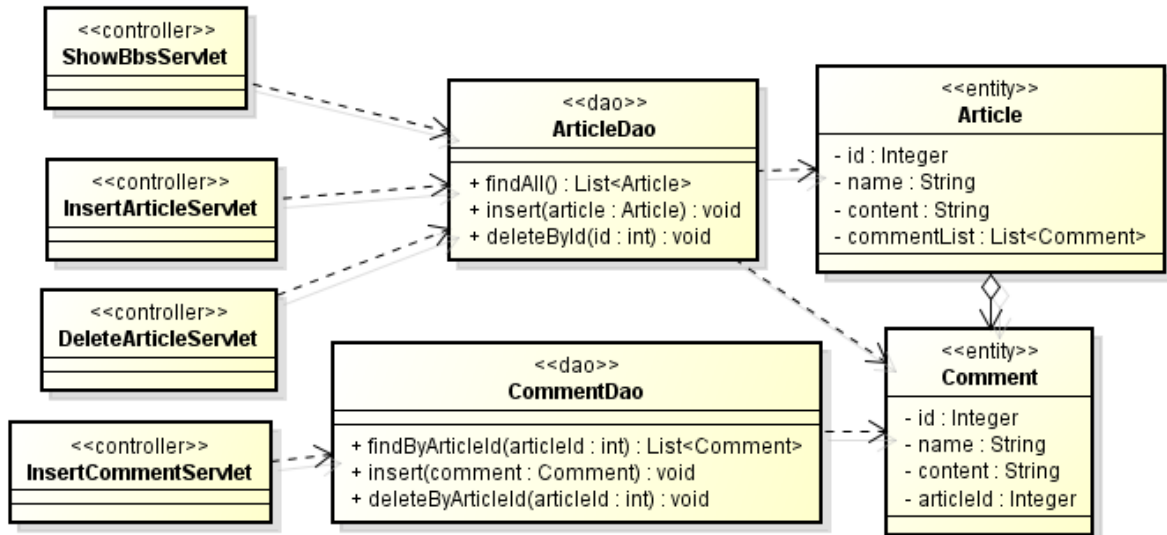
## おすすめの作成順番

- 0.ブラックボックステストのテスト仕様書作成(←1日目午前中)
- 1.記事一覧のみ表示
- 2.記事のみ投稿
- 3.コメントの表示
- 4.コメントの投稿
- 5.記事とコメントの削除
- =====ここまで2日以内が目標(終わったらレビュー依頼)=====
- 6.記事とコメント一覧を結合して表示(中級課題)
- 7.エラーチェック処理(上級課題)
  - 名前と内容が
    - それぞれ空だったら「〇〇を入力して下さい」と出す
    - 名前が50字を超えていたら「名前は50字以内で入力してください」と出す
  - ⇒コメントのエラーは該当箇所のみに出す
- 8.1回のSQLで記事とコメントを一発で消す(上級課題)

## 作成のコツ

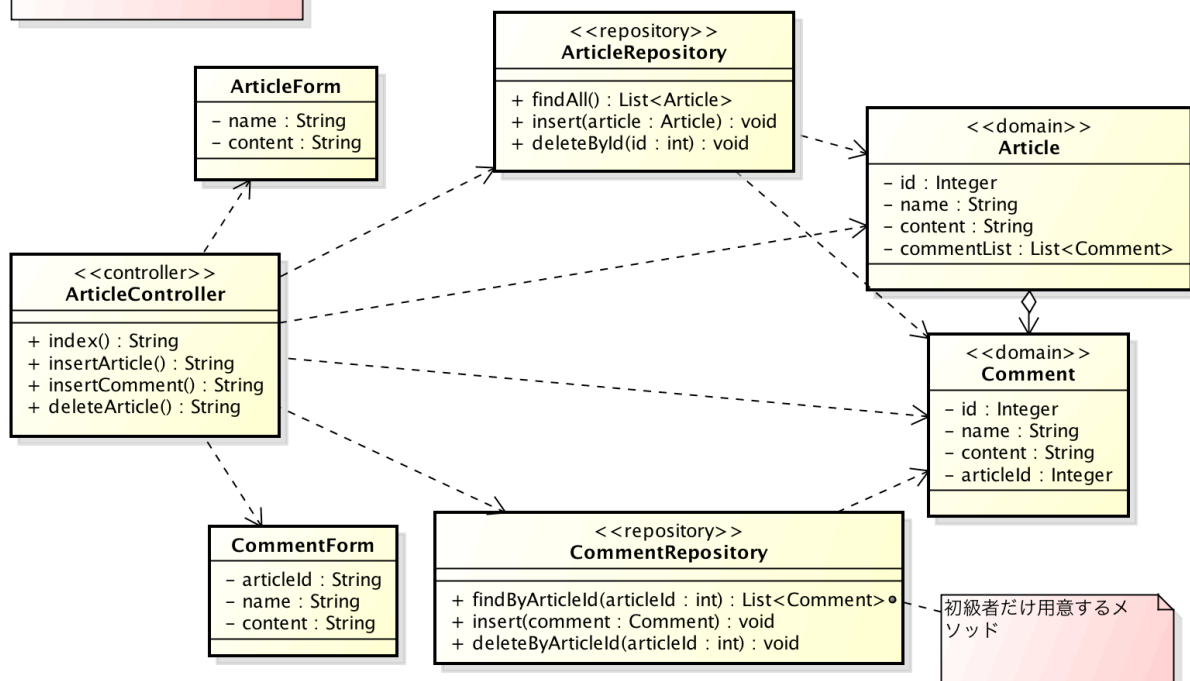
- 開発の進め方
    - 一気に全部作ってから動かす・・・×
    - たくさんのエラーが出て、結局進みが遅くなる
    - 少しずつ動かしながら作成する・・・○
      - 頭をすっきりさせながら作成することができる
  - 作成順序
    - DBに近い順から作成する
    - Domain→Repository→(Service)→Form→Controller→HTML
- ※HTMLに関しては最初に作ってもOKです

## (参考)Servlet組 クラス設計



## Spring組 クラス設計

今回はサービスは作らない方針



## 初級者課題

- 表示時、SQLを発行 × 記事全件検索 1 回 + 記事数
  - ①ArticleRepositoryのfindAll()で全記事を取得
  - ②CommentRepositoryのfindByArticleId()で取ってきた記事一つ一つの全コメントを取得

id	name	content
1	伊賀	伊賀さんの記事
2	山田	山田さんの記事
3	佐藤	佐藤さんの記事

articles

id	name	content	article_id
11	吉田	吉田さんのコメント	1
12	伊藤	伊藤さんのコメント	1
13	山崎	山崎さんのコメント	2

comments

- ①ArticleRepositoryのfindAll()を使って上記の情報を取得
  - ②その後、CommentRepositoryのfindByArticleId()を使って上記の情報を取得

## 中級者課題

- 表示時、SQLを1回発行
  - 結合(JOIN)して検索

id	name	content	com_id	com_name	com_content	article_id
1	伊賀	伊賀さんの記事	11	吉田	吉田さんのコメント	1
1	伊賀	伊賀さんの記事	12	伊藤	伊藤さんのコメント	1
2	山田	山田さんの記事	13	山崎	山崎さんのコメント	2
2	山田	山田さんの記事	14	伊賀	伊賀さんのコメント	2

ヒント

結合すると上記のような結果になります。  
どのタイミングでArticleオブジェクトやCommentオブジェクトを作るか考えましょう。

## 上級者課題

- まずは中級者課題を実施してください
- 追加課題
  - エラーチェック処理（コメントのエラーは該当箇所に出す）
  - 1回のSQLで記事とコメントを一括削除できるように