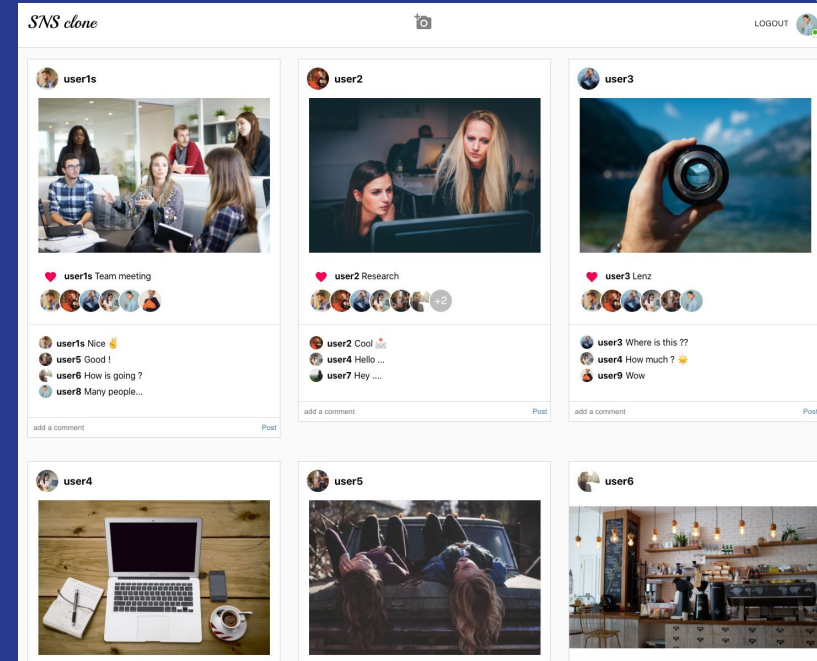
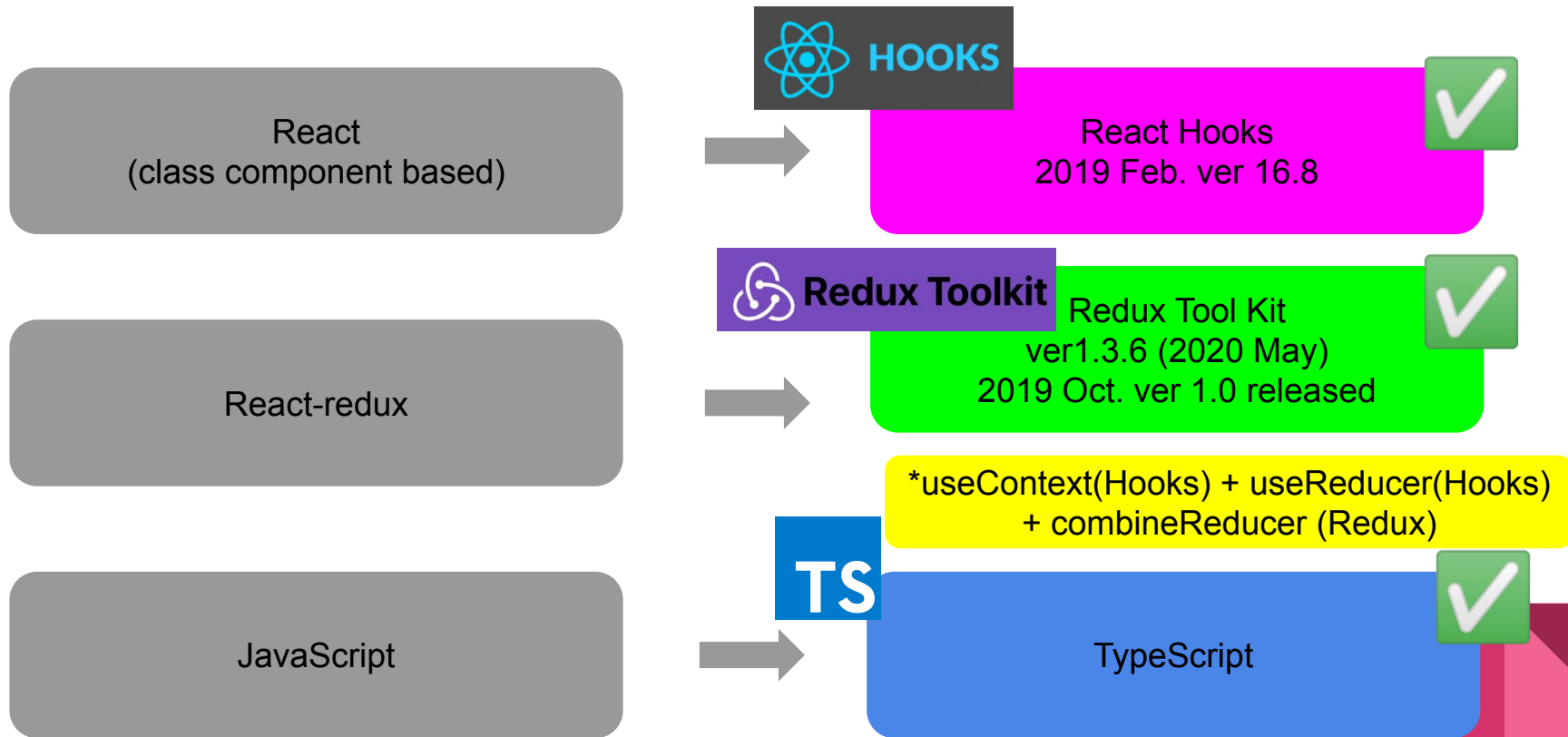


Instagram clone app



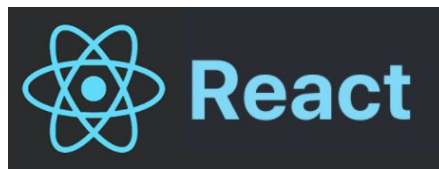
React 開発のトレンド



Instagramで使われている技術



Front End



Back End



Database



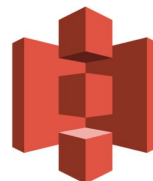
アクティブユーザー
10億人(worldwide)

*Twitter 3億人
(worldwide)

Cloud



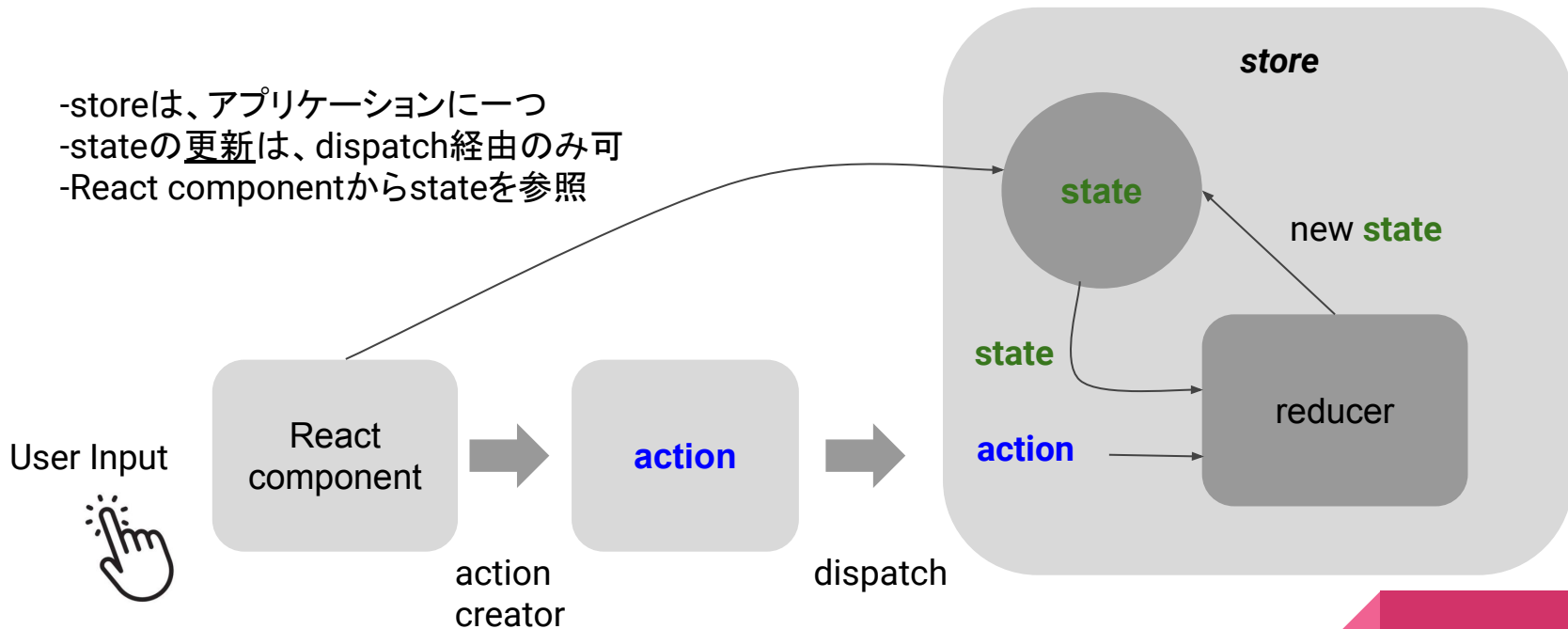
Amazon EC2



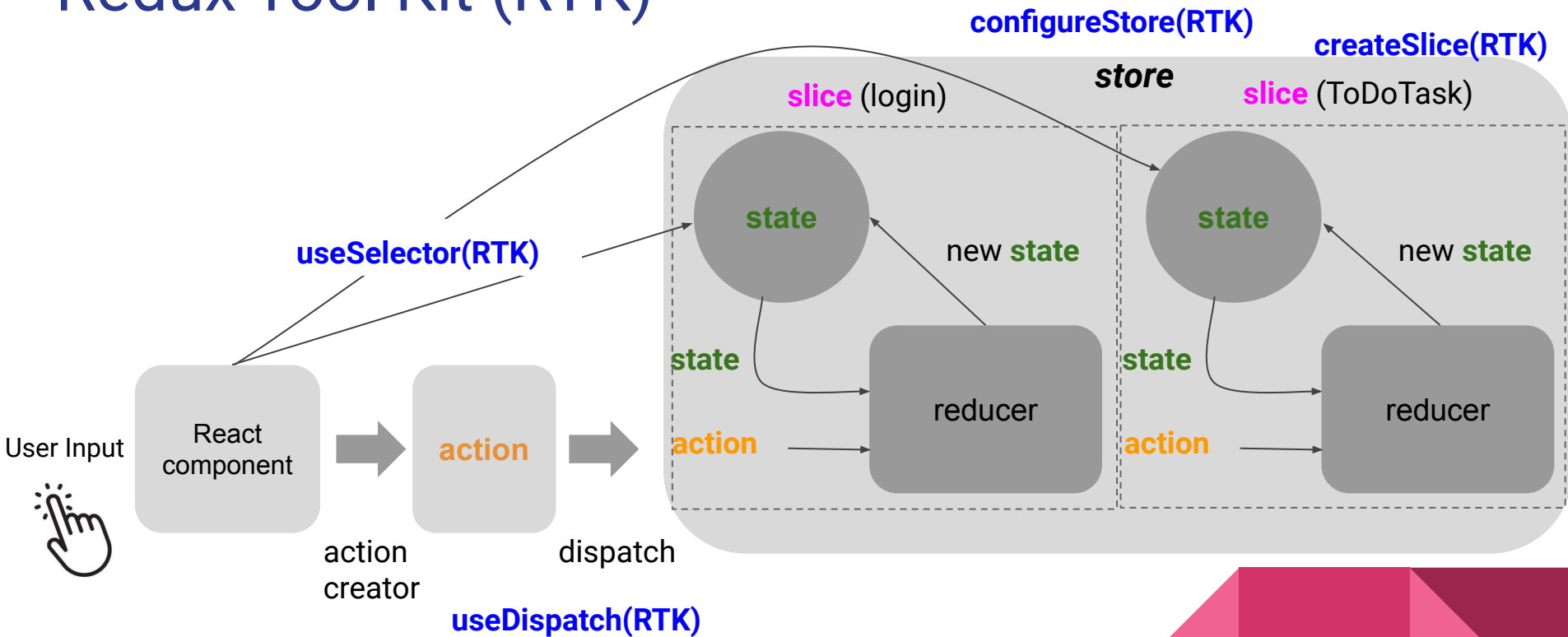
Amazon S3

Reduxの仕組み

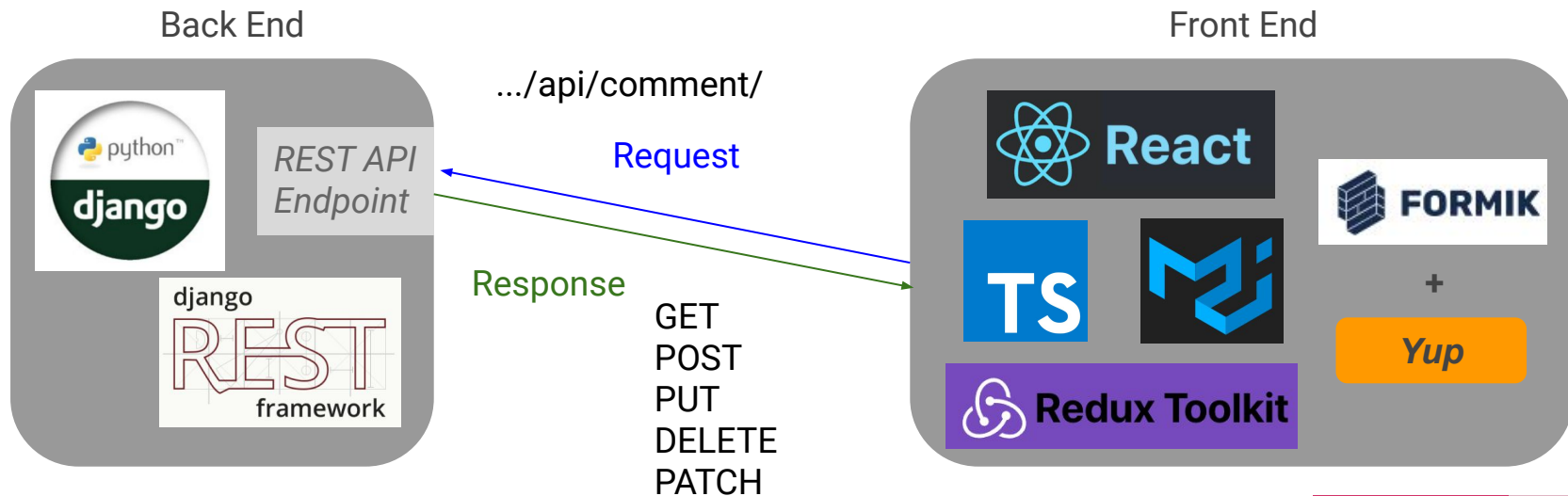
- storeは、アプリケーションに一つ
- stateの更新は、dispatch経由のみ可
- React componentからstateを参照



Redux Tool Kit (RTK)



本コースのハンズオン



Instagram clone App

SNS clone

email

password

LOGIN

[You don't have a account ?](#)

SNS clone



Udemy

LOGOUT




 user1s



❤ user1s Team meeting



 user1s Nice 🍌

 user5 Good !

 user6 How is going ?

 user8 Many people...

add a comment

Post

 user2



❤ user2 Research



 user2 Cool 📧

 user4 Hello ...

 user7 Hey

add a comment


Post

 user3



❤ user3 Lenz



 user3 Where is this ??

 user4 How much ? 🌟

 user9 Wow

add a comment

Post

 user4



 user5



 user6



Modal(ログイン、新規登録)

SNS clone

LOGIN SIGNUP



SNS clone

email

password

LOGIN

[You don't have a account ?](#)


Modal (プロフィール編集)


SNS clone





LOGOUT





User1





User1 Sencond post updated





User1

SNS clone

User1



UPDATE


User1 First comment

DATABASE (モデル構造)

django
REST
framework

User

id
email
password

Profile

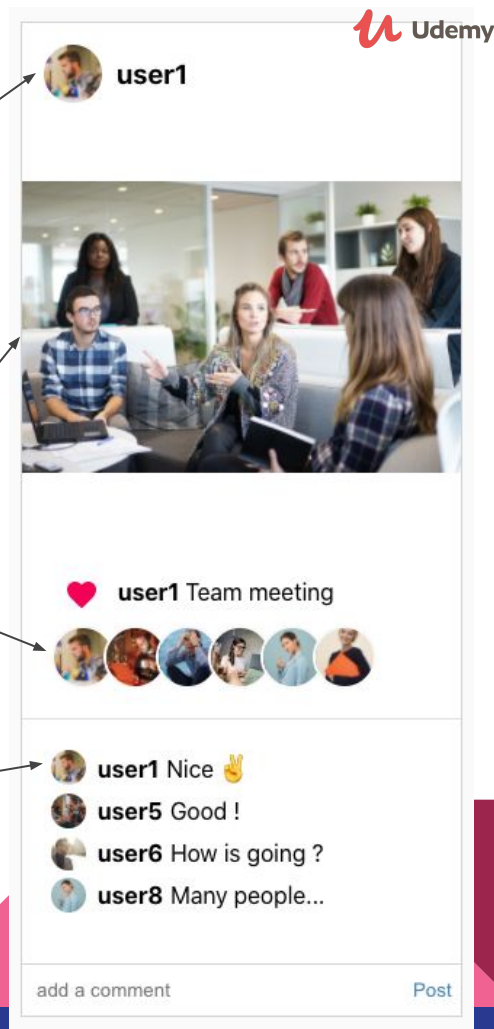
id
nickName
userProfile
img
created_on

Post

id
title
userPost
img
liked
created_on

Comment

id
text
userComment
post



One to One

ユーザーモデル プロフィール

A ↔ A

B ↔ B

C ↔ C

D ↔ D

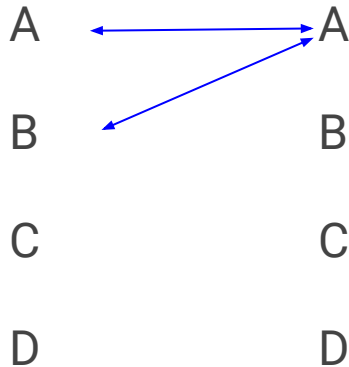
→ **One to One**

One to Many (Foreign key)



コメント

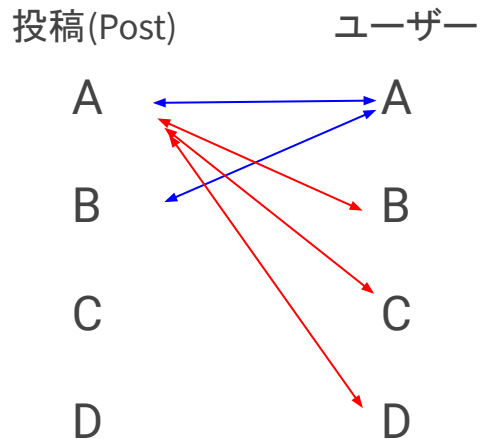
投稿(Post)



各投稿は、複数の Commentを持つ

→ **One to Many**

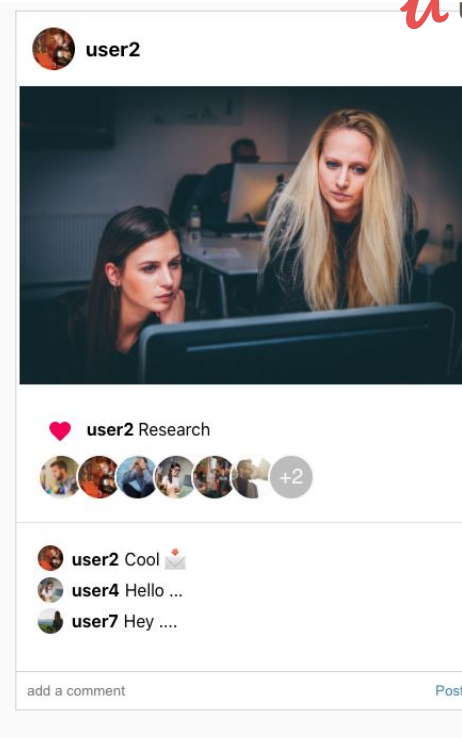
Many to Many



各ユーザーは、複数の投稿にいいね出来る。

+ 各投稿は複数のユーザー (いいねしてくれた)を持つことができる

→ **Many to Many**



API Endpoint (DRF)

api/register/	[POST]	新規アカウント作成
api/myprofile/	[GET]	ログインユーザーのProfile取得
api/comment/	[GET/POST]	コメントの取得と投稿
api/post/	[GET/POST/PUT/PATCH]	投稿の取得、作成、更新、一部更新
api/profile/	[GET/POST/PUT]	プロフィールの取得、作成、更新
authen/jwt/create/	[POST]	JWTトークン取得

Active, Staff, Superuserの違いについて

Active → is_activeに指定するTrue/Falseでアカウントの有効/無効切替

“無効にするとログインできなくなります”

Staff権限 → is_staffに指定するTrue/FalseでStaff権限の有効/無効切替

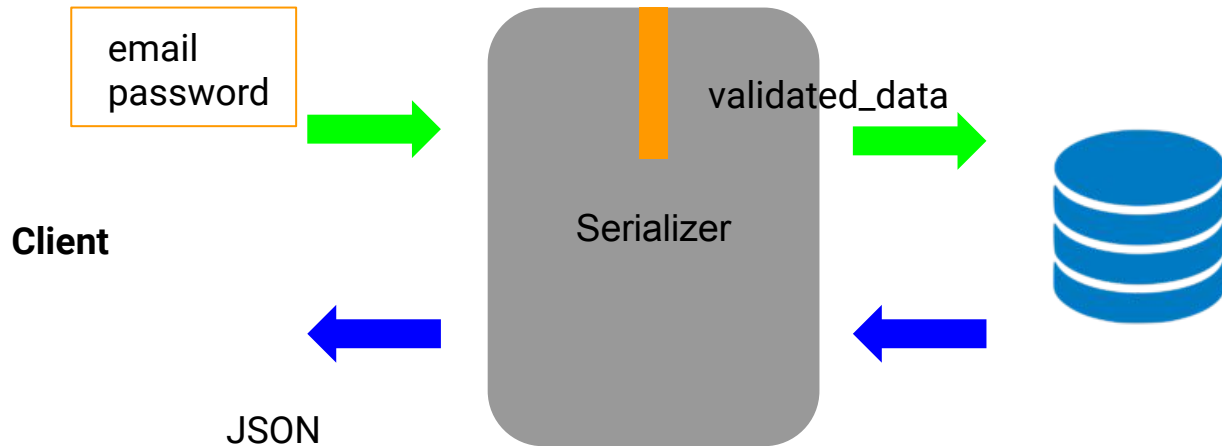
“Staff権限は、AdminのDashboardにログインする権限になります。”

Superuser権限 → is_superuserに指定するTrue/Falseで有効/無効切替

“Superuserは、AdminのDashboardにログインする権限に加えデータベースの内容を変更できるなど全権限を持ちます。”

シリアライザー

例：新規アカウント作成



*特定の属性を渡したくない
*write/read only

ModelViewSet→CRUDを全て使いたい場合は、こちらを使うと便利

単一モデルに対するCRUD(Create, Read, Update, Delete)を標準装備

汎用APIView→CRUDのどれかに特化した機能を使いたい場合に使用

CreateAPIView

ListAPIView

DestroyAPIView

UpdateAPIView

.....



fetchAsyncPatchLiked

