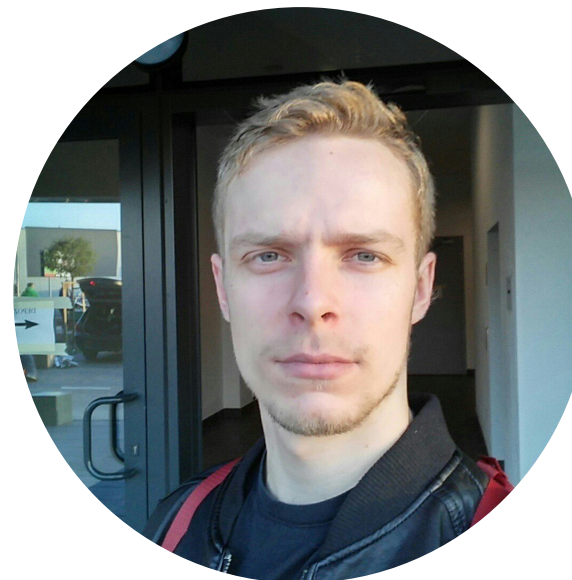


# «React JS»

---

Практика 12.05.19



**Иванов Максим**

преподаватель курса



**LoftSchool**  
от мыслителя к создателю

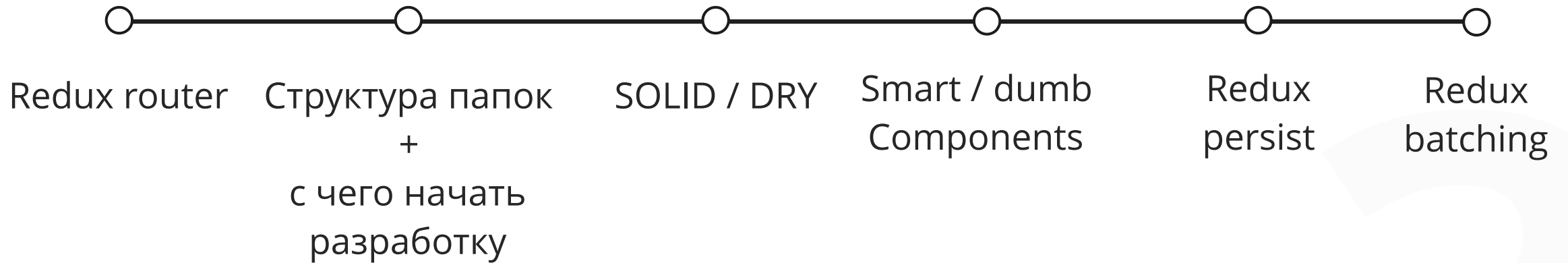
1

# Структура занятия

2

# Структура занятия

---



# Redux-router

---

# Redux router - зачем

---

1. Синхронизировать данные в сторе с роутингом
2. Возможность навигации при диспатче экшенов
3. Поддержка time-travel, дебаг при помощи redux-devtools

# connected-react-router

---

1. <https://github.com/supasate/connected-react-router>

# Структура приложения (с чего начать)

---



app-name

└─ src

└─ page-name

└─ component-name

└─ ... component related files

└─ another-component-name

└─ another-page-name

└─ component-name

└─ ... component related files

└─ another-component-name



**Создание приложения - это во  
многом исследование**

---



# Начните с описания приложения снаружи

---



# Важно продумать структуру данных

---

# Итеративный подход

---



**SOLID / DRY**

---

**LoftSchool**  
от мыслителя к создателю

13

# DRY

---

- Don't
- Repeat
- Yourself

# SOLID

---

- Single responsibility principle
- Open–closed principle
- Liskov substitution principle
- Interface segregation principle
- Dependency inversion principle

# Smart / dumb components

---



# Smart-dumb components

---

- [https://medium.com/@dan\\_abramov/smart-and-dumb-components-7ca2f9a7c7d0](https://medium.com/@dan_abramov/smart-and-dumb-components-7ca2f9a7c7d0)

# Redux persist

---



# Redux-persist

---

- <https://github.com/rt2zz/redux-persist>

# Redux batching

---

# Redux-batching

---

- <https://github.com/tshelburne/redux-batched-actions>
- <https://github.com/manaflair/redux-batch>
- <https://github.com/tappleby/redux-batched-subscribe>

# ВСЕМ СПАСИБО ЗА ВНИМАНИЕ!

---

Добавляйтесь в друзья



[@satansdeer](https://twitter.com/satansdeer)



[youtube.com/user/satansdeer1](https://youtube.com/user/satansdeer1)



[youtube.com/user/loftblog](https://youtube.com/user/loftblog)

**#loftblog #loftschoo**