



Building a Web Server

Yulia Tenincheva

Senior Cloud Engineer, MentorMate

Today's Agenda

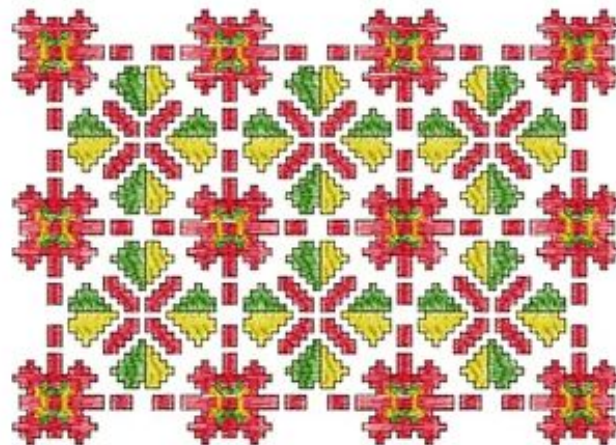
- Software Architectural Patterns
- Popular Node.js Frameworks
- **Getting started** with [Express](#)
- How to approach a new project
- Boilerplates & Generators
- **Middleware** concept
- **Routing**, Data validation



Software Architectural Patterns

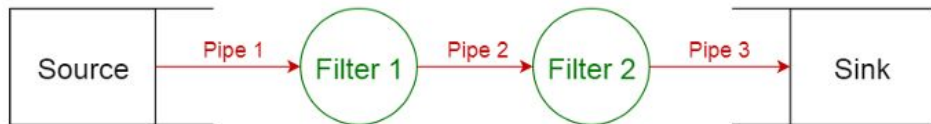
Common Patterns

- Layered
- Client-Server
- Master-slave
- Pipe-filter
- Broker
- Peer-to-peer
- Event bus
- MVC
- Interpreter

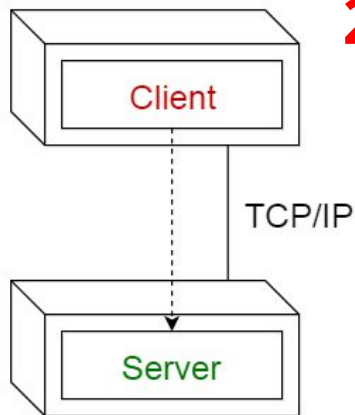


Which pattern is that?

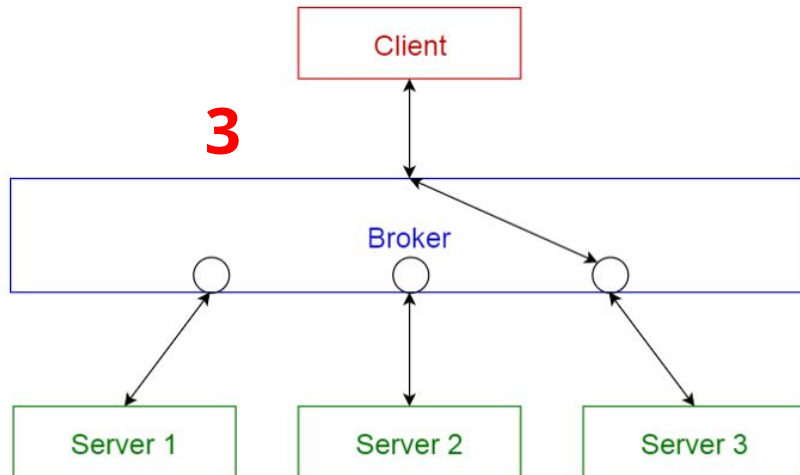
1



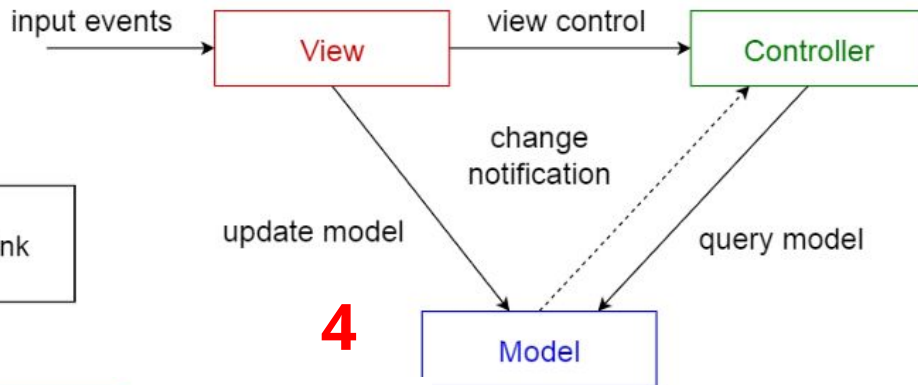
2



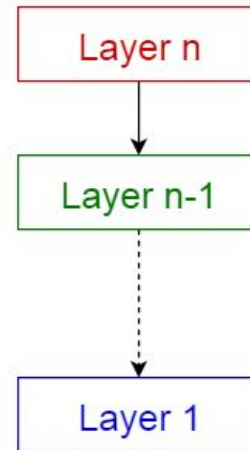
3



4



5



Getting started

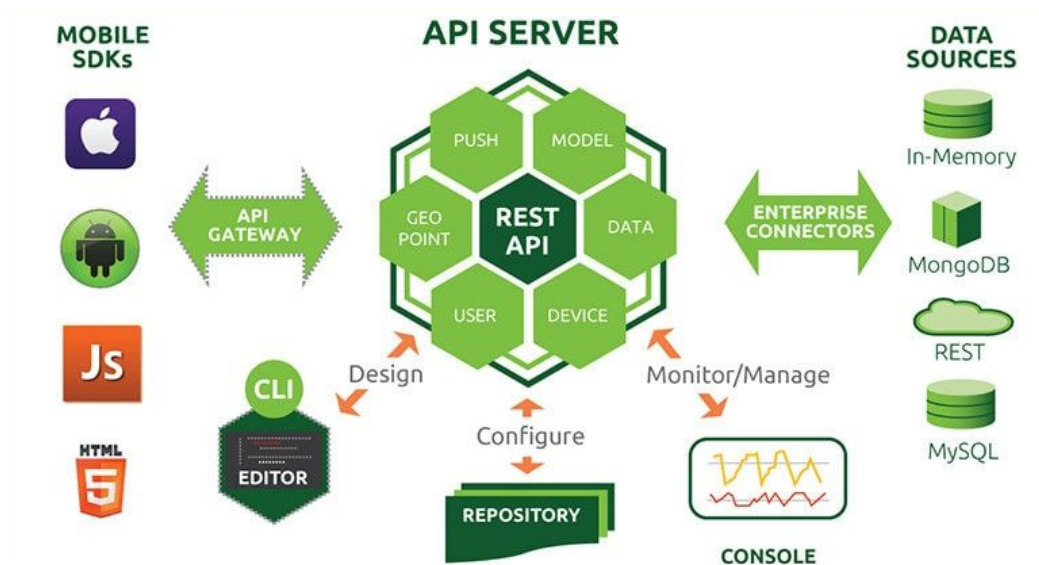
As a developer, consider this

- Platforms, **Infrastructure**
- Choose the **tools** (framework, major libraries, database)
- Define the version control **strategy**, environments, **conventions**
- Prepare the **project structure**
- **Documentation**



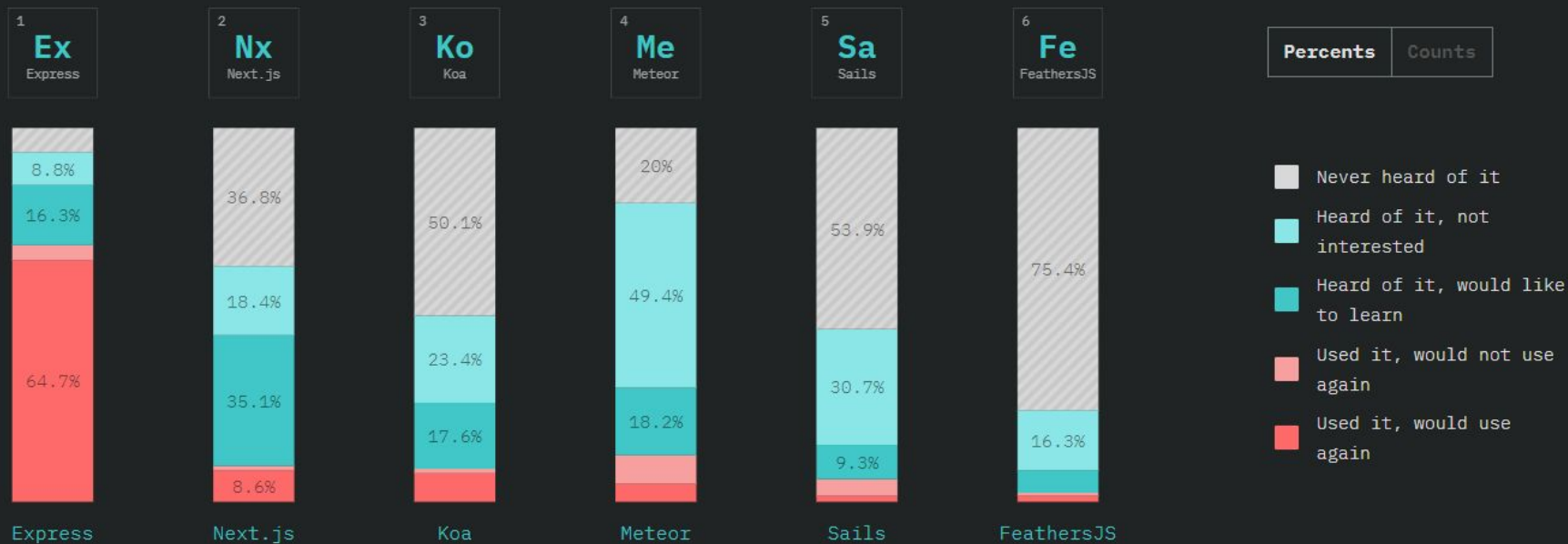
Every project's initial code

- Environment configuration
- Logger
- Authentication / Authorization
- Routes for application health
- Documentation



Node.js Frameworks

Overall Back-end Frameworks results.



Features of Express.js

- Faster development
- [Middleware](#)
- [Routing](#)
- [Templating](#)
- Debugging



Let's Code!

Resources

- What we will build - [Requirements](#)
- Express.js - [API Reference](#)
- REST API Routing Best Practices - [guide](#)
- Adding documentation - [JSDoc](#)
- Query Strings - [Wiki](#), Node.js Module [Docs](#)



Homework
