Express & koa

by Pavel Egorov

Intro (Slide 1)

Hi, everybody! How are you felling?

My name is Pavel Egorov and today I’ll tell you what is it Express and koa, where they are used and the differences between them.

Let’s start!

Slide 2

I’ll start my presentation by telling a few words about Node.js. Anyone know what is Node.js? Node.js is a JavaScript runtime built on [Chrome's V8 JavaScript engine](https://developers.google.com/v8/) that allows to run your JavaScript code outside of browser and allows for one language throughout the entire stack. Node.js provides package ecosystem npm – is the largest ecosystem of open source libraries in the world which included over 280 000 libraries.

Slides 3-5

First topic of my presentation is Express. Anybody heard what is it? Express is Node.js framework for build server side of your web application.

Express includes many different technologies, among them is routing mechanism and middleware.

Express provides access to the MVC (Model-View-Controller) pattern. MVC pattern is a software architectural pattern for implementing user interfaces on computers. It divides an application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user.

Express is asynchronous. Initial release of Express is November 2010. It’s about 7 years ago.

Slides 6-8

Koa… please come in and enjoy the party! =)

What is it? Koa is next generation web framework for Node.js designed by the team behind Express, which aims to be a smaller, more expressive, and more robust foundation for web applications.

Koa doesn’t bundle any middleware within core, and provides an elegant suite of methods that make writing servers fast and enjoyable.

Through leveraging generators Koa will free you from callback hell (let’s say hello to Node.js) and greatly increase error-handling. Koa is more modular then Express. Initial release of koa is 2013. It’s about 4 years ago.

Slides 9, 10

Ok, keep calm and let’s practice! =)

To install Express or koa, first you must install last version of Node.js. Run the Node.js command prompt and create $ npm install express --save for install Express or $ npm install koa --save for install koa.

Slide 11

So, let’s say hello… of course using code! =)

The first step for any developer, when he is working on a Node.js web application, is to create and listening a basic server. So, let’s create a server using each framework to see their similarities and differences. Right away you can see that Koa is similar to Express. Also app.listen() is the exact same function as used in Express.

Express is using the get() method to catch the incoming request of "GET /" and then invoke a callback function that handles two parameters req and res. For this example we are only using res to return back to the page a string using res.send().

The application starts a server and listens of port 3000. It will respond with "Hello, I'm Express (koa)!" after request to the homepage.

Koa is slightly different than Express, it’s using ES6 generators. Any function preceded by a \* means the function will return a generator object.

Slide 12

Ok, let’s see what is generator-functions…

Generator-functions are an ES6/ES2015 feature. Generators are function executions that can be suspended and resumed later.

When you invoke a generator function, it won’t execute the body of the function like a regular function. Instead, it will return a generator object called an iterator. Iterators aren’t a topic of my presentation, but in the context of a generator, it's an object that controls the execution of the generator function via a next() method.

It isn’t until we call gen.next() when the body of generateSequence() starts executing. By calling gen.next(), the generator’s body will execute until the first yield statement and then pause.

Yield-operator is a road in both directions: it not only returns the result to the outside, but it can also transfer the value from the outside to the generator.

Slides 13, 14

That’s not enough. We have to go deeper… =)

Now let’s dig into one of the most important features of a server – routing. Routing – it's a how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, PUT, DELETE and others).

Each route can have one or more handler functions, which are executed when the route is matched.

Some example routes in an application takes the following structure… (point to code).

Slide 15

Now let’s write some middleware and use generators on koa.

When a middleware invokes *yield next* the function suspends and passes control to the next middleware defined. After there are no more middleware to execute downstream, the stack will unwind and each middleware is resumed to perform its upstream behavior.

Slides 16, 17

Each of the current frameworks has its own path.

This is Express way… (point to code). If the current function doesn’t end the request-response cycle, it must call next() to pass control to next function. Otherwise, the request will be left hanging.

Koa way uses generators (point to code). And as I already said, when a middleware invokes *yield next* the function suspends and passes control to next middleware defined. After there are no more middleware to execute downstream, the stack will unwind and each middleware is resumed to perform its upstream behavior.

Slides 18-20

So, let’s see why you should use Express.

Express includes a full feature set what you can use to quickly configure the server and create a web application. The framework has a good documentation, lots of examples and many extensions. Also Express has the largest community of all web application frameworks for Node.js.

Ok, but why is koa better Express?

Koa is more modular, smaller, more robust, more expressive and it makes writing middleware a lot easier than the other frameworks. Koa has better user interface. It's only ES6 framework, for instance its using ES6 generators.

Let’s see on comparison table of Express and koa. If you'd like to be closer to traditional Node.js-style coding, you probably want to use Express. If you want to get rid of callbacks, use Koa.

All two of the current frameworks have a lot in common. All can create a server with just a few lines of code and all make creating a web applications very simple.

Slides 20, 21

If you think that it’s some kind of hell – please ask your questions. =)

Thank you everybody! Bye!