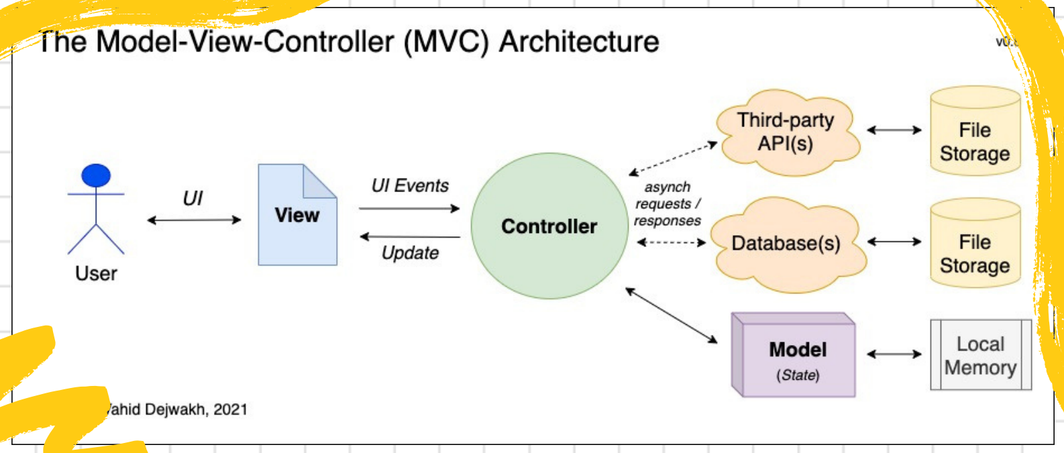
MVC MODEL

MODEL - VIEW - CONTROLLER



INDEX

INTRODUCTION

MVC, knowing how to do it and explain it could be a huge advantage in interviews and your career, we will see step by step what MVC is and how it works, so you can understand the model and apply it to your projects.

WHAT IS MVC?

(AND WHAT IS NOT)

F: It's an architectural paradigm.

G: say what? Is this still 100devs?

F: To make it simple, it's a way to organize and build a project.

G: huh... Keep talking...

F: Well, when we organize concerns on the client side, what should we do?

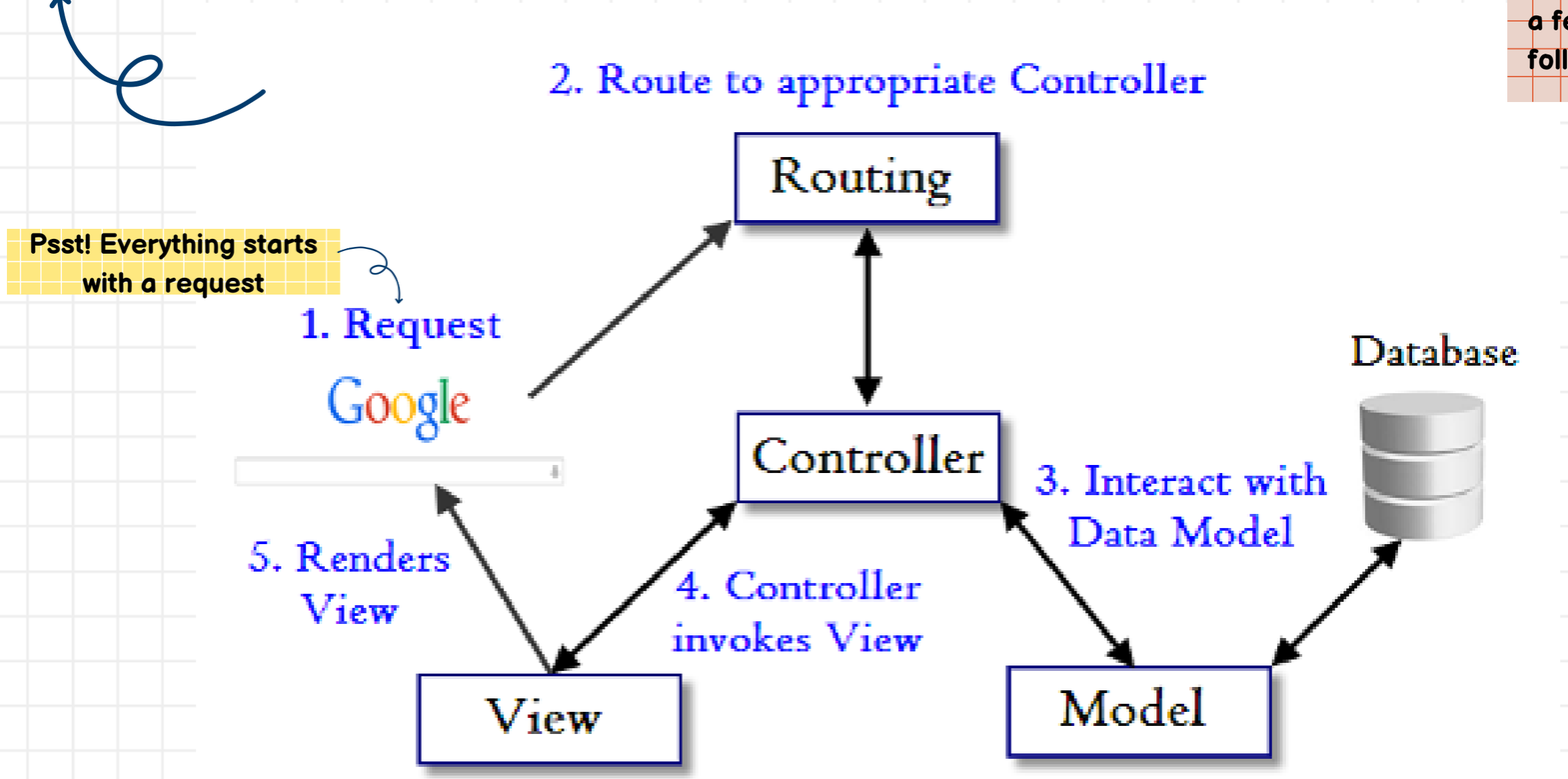
G: If I remember it well, we should separate our work, so we have it right there when we need to edit or correct a mistake.

F: yeah! So it is MVC for the back-end, is a format we use to make a project, and is not a new language or framework you have to learn or library.

I know this doesn't make a lot of sense right now, but I'll explain it better in the next slides

OVERVIEW

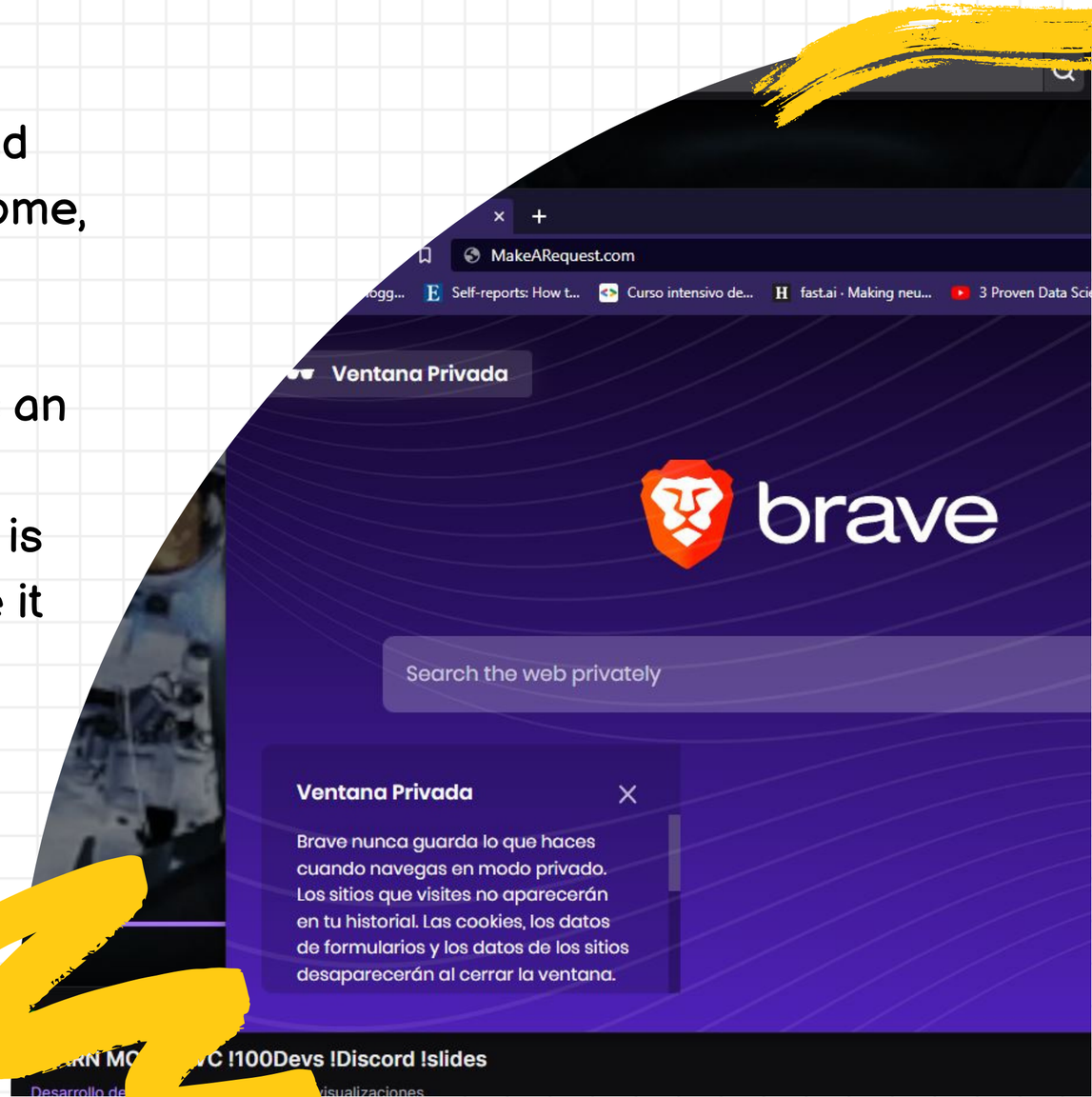
what we should look is the workflow of the model, just spent a few seconds following the steps



01. REQUEST

G: F, I already know the internet are links, and when I type a URL in a browser such as chrome, brave, Firefox, or other, I'll get the page I'm searching for. But... how?

F: well, what you're saying is correct, there is an element you're missing, and is the server. Whenever you make a request, that request is made to a server that is prepared to receive it and respond appropriately with a View.



02. ROUTING

F: Hey G, there's something I didn't tell you... in MVC there's something called routing, so, before the request gets to the controller, we make a routing file where you can find all the routes to each controller method.

G: Ohh, so let me get this straight, follow this flow and tell me if it's ok:

The user sends a request

We search the route of the request

The Route communicates or call a specific controller method

but, can you show me an example?

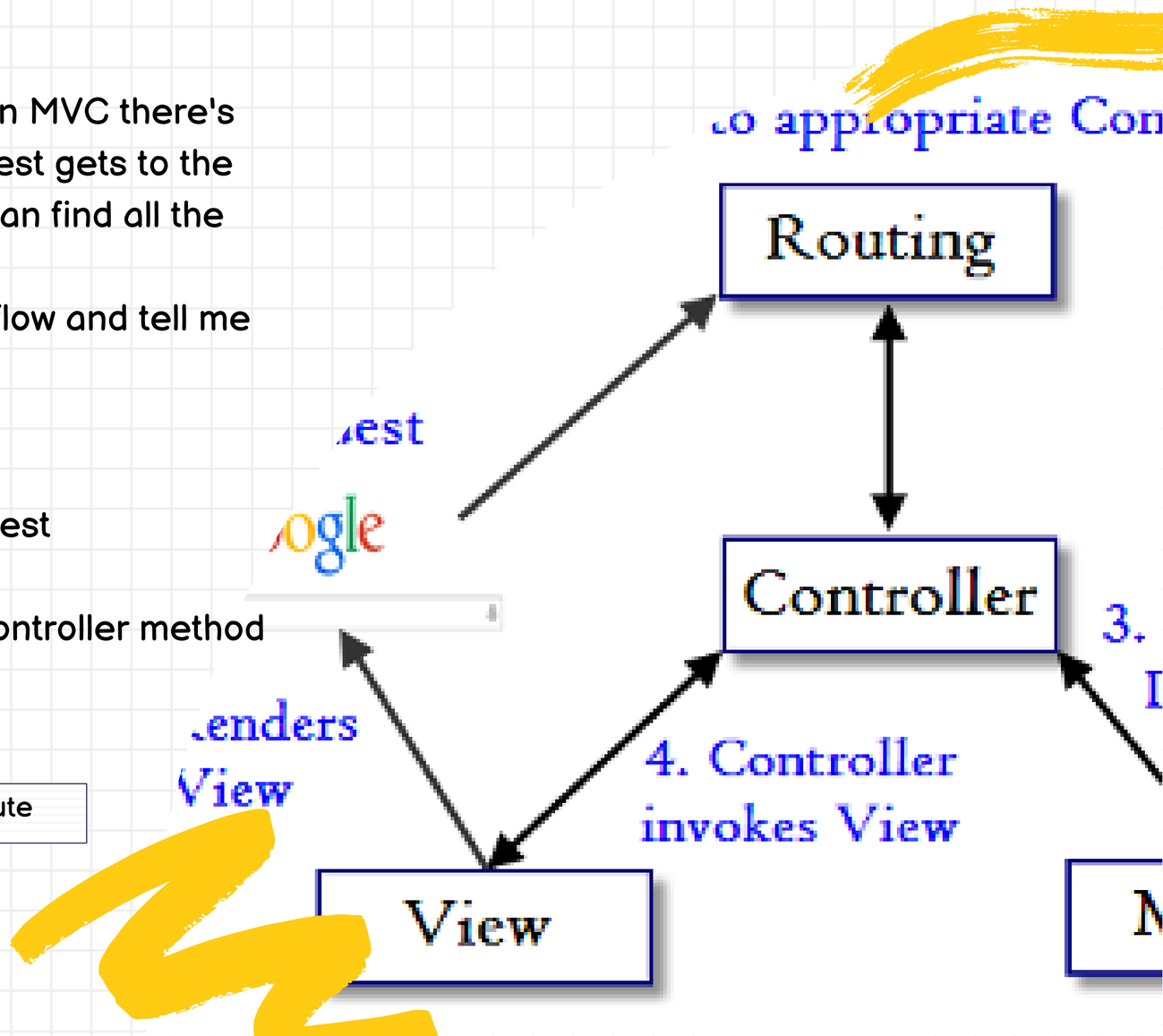
F: Yes, of course, here it is:

code of route

```
const express = require('express')
const router = express.Router()
const todosController = require('../controllers/todos')
const { ensureAuth, ensureGuest } = require('../middleware/auth')

router.get('/', ensureAuth, todosController.getTodos)

router.post('/createTodo', todosController.createTodo)
```



03. CONTROLLER

G: Ok, now I get it, let's keep going, so what the controller does?

F: that's nice, okay the controller after receiving the communication of the route:

*Processes GET/POST/PUT/DESTROY (CRUD) requests

*All server-side logic

*Functions like The Middle Man

-Takes info for user

-Processes info and talks to the DB through the model if needed

-Receives info from the DB through the model

-Speaks to View to explain presentation to the viewer

link to CRUD explanation:

https://www.youtube.com/watch?v=zHq0v5RD_Zk

04. MODEL

G: You mentioned the Model a few times before... what is the model exactly, and what does it do?

F: The model is in charge to communicate with the database, it's rare to see nowadays static websites it's more common to see a web app that needs to store data in a database, these are some of the functions of the model:

*Adding and retrieving items from a database
*Processing data from or to a database
*Speaks with the controller and the controller only

05. DATABASE

F: In the server we don't store data, to store data we use a different structure, we store it in databases such as MongoDB, PostgreSQL, MySQL, etc.

The server does store but only some variables, not all the data we need to make a web app.

Some data we use and get stored in a DB:

*Users

*ID

*Whatever the function of our webApp is a good example is the toDo List, and we use the model to delete or create an item in a DB.

06. VIEW

G: That was quick... so after all this, how do we communicate with our client? Do they need to know this too?

F: No, they don't. We can use React or EJS to make the View, so the user can see what our workflow generates. These are the functions of the View:

*Display what the user sees (UI), Usually consists of HTML/CSS/JS/EJS/React
*Communicates with the controller
*passed dynamic values from the controller
*Template engines.

WHY MVC IS IMPORTANT

5 REASONS TO USE MVC

MVC is used by the most important frameworks such as ruby or Django.

Never repeat yourself by structured your project.

Makes easy the work with a team by having organized files and concerns.

You can save a lot of time using this format. There's no need to see the whole code.

You can have mistakes or errors without break the whole project.

MVC WRAP-UP

MVC is not a language nor a library or data frame, it is a way of doing a project that consists in have our concerns organized. We have three big blocks that communicate to make our web app, that's why this architectural paradigm is called MVC.

M = Model (the one that communicates with DB)

V = View (the one that displays all the answers to our client/user)

C = Controller (the one that communicates with everyone and stores all the methods)

We also have routes, to be more organized, that communicate with our controller.

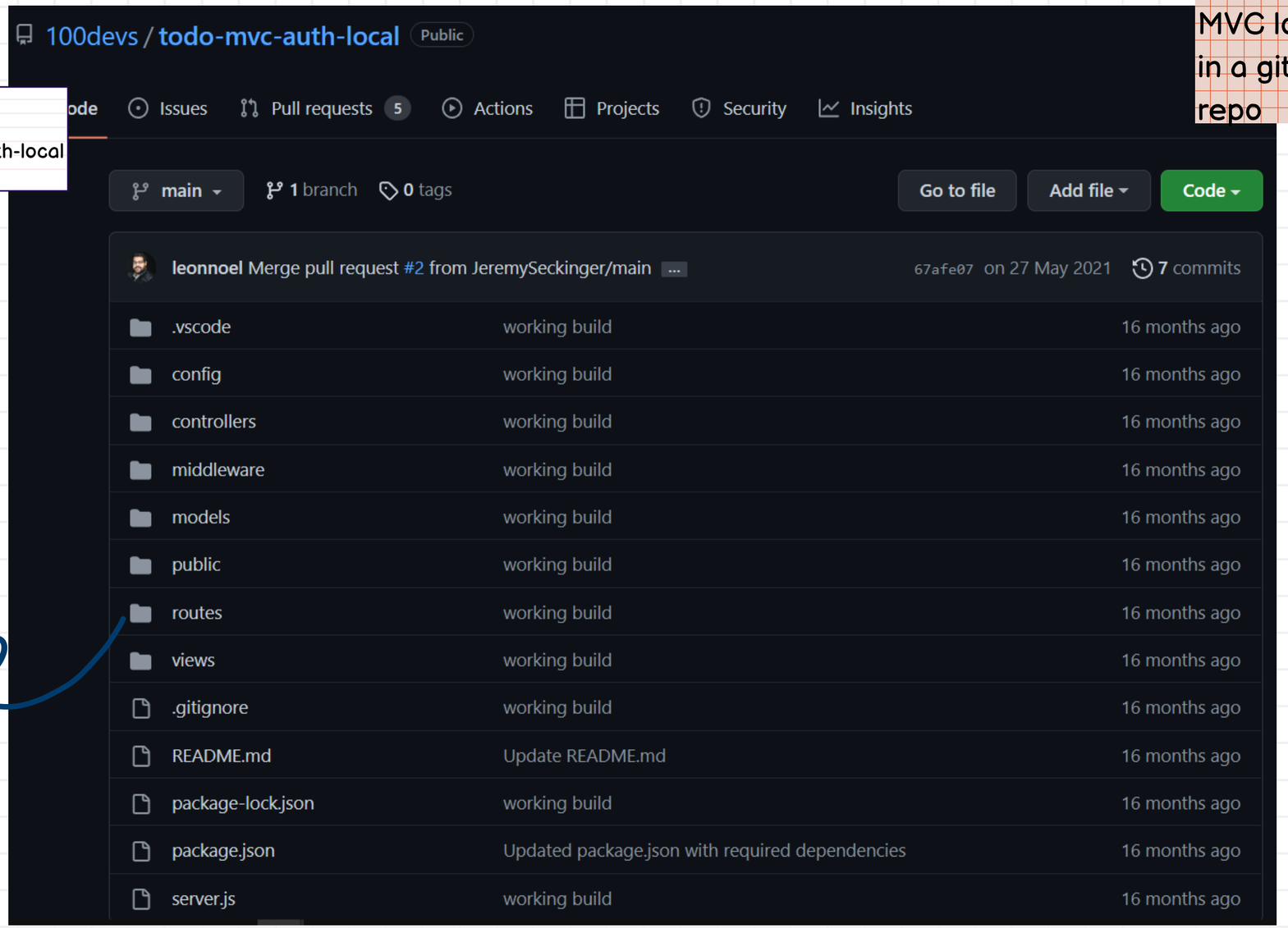
EXAMPLE

link to the repo:

https://github.com/100devs/todo-mvc-auth-local

this is how MVC looks in a github repo

here are our routes :)



THANK YOU FOR READING :)

Fedecha

Twitch: /fedecha_ves
Twitter: /fedecha_ves
Github: /fedechaves
linkedin: /in/fedechavesflecha/