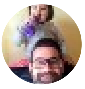

Balint Sera Follow
Jun 15, 2016

On the separation of front-end and backend

Short list of the *separatist arguments*:

1. separate teams with different languages and repos mean faster development cycles (important: javascript devs aren't bound to back-end template languages)
2. separation of concerns: both can concentrate on their goals, architecture, speed, etc. Back-end: data processing, front-end: user interaction
3. *REST* or *GraphQL* API backend is more open to the addition of further clients (native mobile etc)
4. separate deployment: faster server for static frontend resources (see packaging tools like Webpack or Grunt), increased scaleability
5. increased testability

But extra efforts needed:

1. conduct based, well documented API is mandatory to the two team
2. integration tests needed on the front-end

Resources:

Don't Panic: The case for separating front- and back-end

Edit description

dontpanic.42.nl

A Look at Why Some Frontend Developers are Decoupling WordPress

This post was contributed by Emily Miller. A native Hoosier, Emily now lives in San…

wptavern.com

Nick Sergeant

Edit description

nicksergeant.com

Six Reasons We Split Front-end and Back-end Code Into Two Git Repositories |…

Recently, a client of ours asked us why our developers split front-end (client) and back…

quickleft.com