

最速でJavaエンジニア になって稼ぐ

2018/11/17

人生逃げ切りオンラインサロン内

目次的なやつ

- 出来るエンジニアの仕事の進め方
- アプリケーションを作る時のちょっとしたコツ
- 作成するアプリケーションの要件・仕様
- バージョン管理(Git)
- ORMapper(MyBatis)

出来るエンジニアの仕事の進め方

- 現在エンジニアとして仕事をしている人は「技術」に関するすべてを知った上で仕事をしているわけではない(当たり前のことだけど)
- 仕事においては「わからない事」「知らない事」「やったことがない事」というような「未知の領域」に対して「結果を出す事」が求められることが殆ど。
- 募集要項に書かれている経験年数や習得技術はあくまで「前提」でしかない。
- この「未知」に対して「どういうアプローチ」を取って「結果を出す」か。これがエンジニア(個人)によって全く違う仕事の進め方になる。
- しかし、全く違うとはいえ「ある程度の結果を出す人」の行動パターンみたいなものはある。それを一部抜粋。

「無理」とは思わず
「出来る」前提で考える。

とにかくググる。

めちゃくちゃ質問する。

でも人の言った事を「鵜呑み」にしない。

色々なパターンを考える。

手を動かして検証する。

そしてまたググる。

「出来なかった」時のことを考える。

「ごめんなさい」が最速。

というわけで

今回からはちょっと講座内容的にも
「突き放し気味」になります。

「自己解決能力」を身に付けていきま
しょう。

アプリケーションを作る時の
ちょっとしたコツ

画面 ⇔ プログラム ⇔ DB

お客さんが気にするのは見た目。

プログラムは中間。

要件や仕様を満たすための
「データの構造」を表現するのがDB。

中間から進めるのは要領が悪い。
なので、画面 or DBから作る。

作成するアプリケーションの 要件・仕様

- 業務日付(翌月末、月末最終日)のようなものを管理したい。
- 業務日付の計算式を登録して、計算基準日を元に計算したい。
- 計算は画面上から手動でできればとりあえず良い。
- 業務日付は「他の機能」でも使いたい為、任意の日付IDみたいなもので検索して取得できるようにしてほしい。
- 検索で取得する際にはサービス経由での検索か、WebAPIのような形で検索したい。通信フォーマットはJSONでやり取りしたい。
- どんな業務日付が登録されているかも確認したいので、一覧取得も出来るようにしたい。これもWebAPIの形式でやりたい。

ヒント1

INPUT

画面

計算基準日

API(計算)

日付ID

計算基準日

API(一覧取得)

なし

OUTPUT

画面

No.

日付ID

日付名

計算結果

計算式(年月日)

API(計算)

計算結果

API(一覧取得)

日付ID

日付名

計算式

ヒント2

APIは後回し。
まずは画面上で一覧取得とか計算ができるように。

ヒント3

DBのテーブルは1個でいける(多分)

ヒント4

画面とテーブルの要素は一致してても
良い。

ヒント5

出来るエンジニアの仕事の進め方を
参照する。

バージョン管理(Git)

- プログラムを書く際、何度も修正をするし、新しいコードを書いたり、削除したりもする。
- この時に「常に上書き保存」というのがめちゃくちゃ危ない。
- というのも「既に動いているプログラム」を修正するという事は「動かなくなる」リスクを常に秘めている。これはどれだけ優秀な人でも起こる。
- なら「起こる」事を前提にして、それでも大丈夫、という状態にしておきたい。
- それが「バージョン管理システム」。
- 保存した内容を常に累積しておき、以前の状態へすぐに戻すことが可能。
- 大きく分けるなら「Git」が「SVN」が主流。Gitが最近は人気。

今回使うのはGitベースのGitHub。

GitHubに登録して、アプリをバージョン管理してみよう。

Gitの解説、GitHubの登録からバージョン管理まで
<https://tech-camp.in/note/technology/4938/>

ORMapper(MyBatis)

プログラム ⇔ DB
のやり取りを楽しむ為のもの。

- オブジェクト関係マッピングの略。
- Javaなどのプログラムコードで書かれた「オブジェクト」を「DBの構造体」にマッピングする仕組み。
- ぶっちゃけ実例を見ないとわからない + プログラムに慣れてないとよくわからない。
- というような代物なので、色々試してみるのが吉。
- 現場では当たり前のように使うので、使えないと辛い。

SpringBoot + MyBatisのサンプルコードや使い方など
<https://www.shookuro.com/entry/2017/11/23/203318>

上記よりもちょっと踏み込んだサンプル
<https://qiita.com/kazuki43zoo/items/ea79e206d7c2e990e478>
<https://qiita.com/TEBASAKI/items/8925aa675f5e15b41967>

次回までの宿題

検索が出来たら、登録・更新。
それも出来たらAPI実装に進んでみよう。

おしまい