

Informatika pro moderní fyziky (11)

formát JSON, použití cizích API; zadání zápočtových úloh

František HAVLŮJ

e-mail: haf@ujv.cz

ÚJV Řež

oddělení Reaktorové fyziky a podpory palivového cyklu

akademický rok 2015/2016

22. prosince 2014

- 1 K zápočtovým úlohám
- 2 Persistence dat a formát JSON
- 3 Použití cizích API

Obsah

- 1 K zápočtovým úlohám
- 2 Persistence dat a formát JSON
- 3 Použití cizích API

Oobecně:

- ke každému zadání jsou k dispozici vzorová data
- já to budu testovat i na datech jiných
- očekávám, že všechno proběhne na jedno spuštění skriptu / rake tasku
- každý má k dispozici jeden pokus řádný a jeden opravný

Klasifikace

- F - nejde to spustit, ani pro zadaná data to v podstatných bodech nesplňuje zadání
- E - pro zadaná data to funguje, ale pro jiná čísla to nechodí
- D - obecně to funguje, ale stejně chybí drobnosti ze zadání
- C - všechno funguje jak má
- B - funguje a navíc jsou výstupy hezké a přehledné, soubory nejsou generovány “na velkou hromadu”, ale roztrženy do složek apod.
- A - kromě výše uvedeného jsou splněny i požadavky formy (správné odsazování, rozumná jména funkcí a proměnných) a efektivity (je to rozumně naprogramované - vhodné použití funkcí, datových struktur atd.)

Známka se snižuje o stupeň, pokud:

- jsou někde ve skriptech použity absolutní cesty, takže je budu muset upravovat (výjimkou jsou cesty k programům jako např. gnuplot, které ovšem musí být umístěny v proměnné někde na začátku skriptu (abych to nemusel lovit)
- bude v kódu něco, co limituje použití na OS Windows (backslash v cestě, kódování win1250 atd.)

(a podobně)

Obsah

- 1 K zápočtovým úlohám
- 2 Persistence dat a formát JSON**
- 3 Použití cizích API

Ukládání strukturovaných dat

- často mám data ve formě struktury (kombinace hash+pole, různý stupeň vnoření)
- z různých důvodů můžu chtít data uložit na disk a pak je znovu načítat
- (zejména efektivita zpracování, případně data z externích/webových zdrojů)
- bylo by dobré mít možnost uložit a načíst rovnou celý hash
- odpověď jsou strukturované metaformáty - YAML, XML, JSON

Práce s JSON

- v Ruby je k mání knihovna – `require 'json'`
- generování JSON: `hash.to_json`
- čtení JSON: `JSON[data]`

Příklad – výsledky běhu kolem rybníka

- dva soubory – `ages.csv`, `times.csv`
- chci v jednom skriptu (tasku) načíst, spárovat a uložit
- a v jiném už rovnou načíst zpracovaná data
- (viz minule, metoda `load_data`)
- pozor! JSON nezná symboly, uloží se jako řetězce

Obsah

- 1 K zápočtovým úlohám
- 2 Persistence dat a formát JSON
- 3 Použití cizích API**

K čemu to?

- spousta informací na webu je poskytována ve strojově čitelné formě
- API – rozhraní mezi aplikacemi
- s využitím webových služeb naše možnosti exponenciálně rostou (počasí, doprava, mapy, atd atd.)
- spousta věcí se dá udělat jako *mashup* – sice nic neumím, ale umím to dát dohromady

Typy / formáty

- URL – rovnou dostanu např. obrázek po zadání správného URL
- XML – velmi obecný, ale komplikovaný formát (vypadá jako HTML)
- JSON – velmi jednoduchý a kompaktní formát, vyvinutý pro JS (v podstatě jen číslo, řetězec, pole, hash)

URL API – google maps

- stačí správně vymyslet
- pozor na usage limits (v produkci je nutné lokální cache...)
- QR platba:

`http://qr-platba.cz/pro-vyvojare/restful-api/#generator-image`

- Google Maps static API:

`https://developers.google.com/maps/documentation/staticmaps/`

Jednoduchý mashup: mapa o-závodů

- ORIS API – <http://oris.orientacnisporty.cz/API>
- úkol: vypišme kalendář MTBO závodů v roce 2016
- `http://oris.orientacnisporty.cz/API/?format=json&method=getEventList
&sport=3&datefrom=2015-01-01&dateto=2015-12-31 ...`
- <https://developers.google.com/maps/documentation/javascript/examples/map-simple>
- API klíč – v praxi si musíte pořídit vlastní (ale nic to nestojí)
- AlzaSyADJv1v8oj7ePv6q8H8he1zwbZtXgrfcek

A to je vše, přátelé!

