Standard Model Database App Documentation

Stack used

Node.js/Express.js
MongoDB
Jade for templating
jQuery / CoffeeScript
Mocha for unit testing

Installation

MongoDB

no user nor password needed
$mongo
> use standardModel
> load("path to project dir/db.js") to load data

Unit tests

$ mocha

Starting

$ node bin/www

MongoDB

I started with 3 separate collections (particles, physicists, particle groups), but after struggling with 'joining' them in code, I eventually sticked to denormalized, flat model, which apparently is a standard approach anyway.

Notice there is no search index, it would be obviously needed for higher volume of database documents.

Code

The code was generated using express generator with default settings (like Jade for templating).

Spelling correction

I used a naive approach with Levenhstein distance being calculated for every particle name in database and name with minimal distance as a suggestion is returned. Iterating over all names makes it probably a little bit of ineffective. It could be tweaked more, e.g. for the moment searching for 'tao' or 'qqq' both suggest 'tau', since both are Levenhstein distance values of 3.

CoffeeScript use

Just client-side code is written in CS (i.e. textbox watermarking and a few more)

Grid displaying

I made it easy and didn't bother with configuring it just with CSS. jQuery on load is used to set main div width, so exactly 3 particle divs fit in one row.