Upgrading Zeus

This document describes the main steps taken to upgrade the Zeus app, based on the prior analysis detailed in [Review of Zeus Software Upgrades](https://docs.google.com/a/consected.com/document/d/1oAWjCNgczfsnRaxczzGPeGWyZ8H8P-fyT0-30RNF2g8/edit?usp=sharing).

## Rails

Upgrade from **4.2.5.1** to **4.2.9**

Create a sample Rails app with version 4.2.9 in order to view updates required to the Gemfile, since certain gem versions are specified in the current Gemfile that may need to be edited in the file to ensure correct update.

|  |  |
| --- | --- |
|  | mkdir rails429 |
|  | cd rails429 |
|  |  |
|  | # install Rails 4.2.9 |
|  | echo 'source "https://rubygems.org"' > Gemfile |
|  | echo 'gem "rails", "4.2.9"' >> Gemfile |
|  | bundle install |
|  |  |
|  | # create a sample Rails 4.2.9 app, with Postgres as the DB |
|  | bundle exec rails new app429 -d postgresql |

Now review the Gemfile created for *app429* comparing it to the existing Gemfile. Do this by eye rather than diff, since there are additions we made for a reason.

|  |  |  |
| --- | --- | --- |
| Zeus Gemfile | app429 (sample) Gemfile | Notes |
| source 'https://rubygems.org'  gem 'rails', '4.2.5.1'  gem 'pg'    # Forcing this version, as rake 10.5 breaks the build  **gem 'rake', '10.4.2'**  gem 'jquery-rails'    gem 'devise'  gem 'dalli'  gem 'country\_select'  gem 'syslog-logger'  group :doc do  gem 'sdoc', '~> 0.4.0', group: :doc  end      group :development, :test do  # Call 'byebug' anywhere in the code to stop execution and get a debugger console  gem 'byebug'    # Spring speeds up development by  keeping your application running in the background. Read more: https://github.com/rails/spring  gem 'spring'  gem 'rspec-rails', '~> 3.0'  gem 'webrick'  gem "brakeman", :require => false  gem "bundler-audit"    gem 'sass-rails', '~> 5.0'  gem 'uglifier', '>= 1.3.0'  end    group :test do  gem 'capybara'  gem 'selenium-webdriver'  gem 'simplecov', :require => false  end    group :production do  #gem 'passenger'  end | source 'https://rubygems.org'  # Bundle edge Rails instead: gem 'rails', github: 'rails/rails'  gem 'rails', **'4.2.9'**  # Use postgresql as the database for Active Record  gem 'pg', **'~> 0.15'**  # Use SCSS for stylesheets  gem 'sass-rails', '~> 5.0'  # Use Uglifier as compressor for JavaScript assets  gem 'uglifier', '>= 1.3.0'  # Use CoffeeScript for .coffee assets and views  gem 'coffee-rails', '~> 4.1.0'  # See https://github.com/rails/execjs#readme for more supported runtimes  # gem 'therubyracer', platforms: :ruby    # Use jquery as the JavaScript library  gem 'jquery-rails'  # Turbolinks makes following links in your web application faster. Read more: https://github.com/rails/turbolinks  gem 'turbolinks'  # Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder  gem 'jbuilder', '~> 2.0'  # bundle exec rake doc:rails generates the API under doc/api.  gem 'sdoc', '~> 0.4.0', group: :doc  # Use ActiveModel has\_secure\_password  # gem 'bcrypt', '~> 3.1.7'  # Use Unicorn as the app server  # gem 'unicorn'  # Use Capistrano for deployment  # gem 'capistrano-rails', group: :development    group :development, :test do  # Call 'byebug' anywhere in the code to stop execution and get a debugger console  gem 'byebug'  end    group :development do  # Access an IRB console on exception pages or by using <%= console %> in views  gem 'web-console', '~> 2.0'    # Spring speeds up development by keeping your application running in the background. Read more: https://github.com/rails/spring  gem 'spring'  end | Make this change  Make this change to be sure  These gems are included only in development and test block below in Zeus, since we compile Javascript and CSS assets in the staged build environment, not on the production server  This is not included in Zeus, since we do not use CoffeeScript  Rake version was forced in Zeus, since there was a recognized bug that prevented build in certain environments. We should try to remove this directive now to allow standard Rake version to be installed  Turbolinks is not used in Zeus  Devise, dalli, country\_select and syslog-logger are third-party gems used in Zeus.  Jbuilder is not used in Zeus  Zeus just limited where this is installed  web-console not used in development currently. Reasonable to install this as it is a useful development tool that does not affect the production server  Rspec is used for testing Webrick is used exclusively in development and test  Brakeman provides source code security analysis  Bundler audit provides gem security analysis  Sass-rails and uglifier match the new versions  These are test specific gems  Placeholder that reminds us that the Passenger app server is deployed through the Ansible role. |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|  |  |
|  |  |
|  |  |

Run **bundle update** to bring the Gems up to date according to the Gemfile.

Run **rspec** to run the automated tests. We’ll step through each of the errors in turn.

**#1**

WARN: Unresolved specs during Gem::Specification.reset:

diff-lcs (< 2.0, >= 1.2.0)

WARN: Clearing out unresolved specs.

Please report a bug if this causes problems.

This appears to be a development environment issue. Cleaning up the gem solved the issue:

gem cleanup diff-lcs

**#2**

The PGconn, PGresult, and PGError constants are deprecated, and will be

removed as of version 1.0.

You should use PG::Connection, PG::Result, and PG::Error instead, respectively.

Called from /home/phil/.rbenv/versions/2.2.4/lib/ruby/gems/2.2.0/gems/activerecord-4.2.9/lib/active\_record/connection\_adapters/postgresql\_adapter.rb:44:in `new'

This is a message from the **pg** gem about deprecation of constants that appear within Rails ActiveRecord. The Zeus source code does not include any of the affected constants, and it is expected that Rails will fix this before there is actually an issue.

**#3**

DEPRECATION WARNING: [Devise] including `Devise::TestHelpers` is deprecated and will be removed from Devise.

For controller tests, please include `Devise::Test::ControllerHelpers` instead.

(called from <top (required)> at /home/phil/NetBeansProjects/fpa-phase3/spec/controllers/addresses\_controller\_spec.rb:3)

DEPRECATION WARNING: [Devise] including `Devise::TestHelpers` is deprecated and will be removed from Devise.

For controller tests, please include `Devise::Test::ControllerHelpers` instead.

(called from <top (required)> at /home/phil/NetBeansProjects/fpa-phase3/spec/controllers/admin/accuracy\_scores\_controller\_spec.rb:4)

… and more ...

These deprecation warnings were fixed in **spec/rails\_helper.rb**

# removed Devise::TestHelpers from the following line, since it is now deprecated.

# Using Devise::Test::ControllerHelpers as advised

config.include Devise::Test::ControllerHelpers, type: :controller

**#4**

Failures:

1) Admin::AccuracyScoresController behaves like a standard admin controller GET #show assigns the requested item as @var

Failure/Error: expect(show: {:id => item\_id}).not\_to be\_routable

NoMethodError:

undefined method `encoding' for {:id=>"3697"}:Hash

Shared Example Group: "a standard admin controller" called from ./spec/controllers/admin/accuracy\_scores\_controller\_spec.rb:19

# ./spec/support/standard\_admin\_controller.rb:58:in `block (3 levels) in <top (required)>'

2) Admin::CollegesController behaves like a standard admin controller GET #show assigns the requested item as @var

Failure/Error: expect(show: {:id => item\_id}).not\_to be\_routable

NoMethodError:

undefined method `encoding' for {:id=>"1643"}:Hash

Shared Example Group: "a standard admin controller" called from ./spec/controllers/admin/colleges\_controller\_spec.rb:26

# ./spec/support/standard\_admin\_controller.rb:58:in `block (3 levels) in <top (required)>'

… and more...

This appears to be due to a change in rspec controller. Either:

* This mechanism for checking whether a path was invalid is now broken, or
* The test was actually invalid but was passing, and this hole has been fixed by rspec

Either way, fixing up the general rspec test, which was used in many places, resolves the issue. So editing:

./spec/support/standard\_admin\_controller.rb:58

From:

expect(show: {:id => item\_id}).not\_to be\_routable

To:

expect(get: "#{path\_prefix}/#{object\_symbol}/#{item\_id}" ).not\_to be\_routable

Similar changes were made in:

./spec/admin/manage\_users\_controller\_spec.rb

./spec/item\_flags\_controller\_spec.rb

./spec/pro\_infos\_controller\_spec.rb

./spec/protocol\_events\_controller\_spec.rb

./spec/sub\_processes\_controller\_spec.rb

**#5**

Continuing with the subsequent errors of type:

1) admin sign in process should prevent invalid sign in

Failure/Error: expect(page).to have\_css ".flash .alert", text: "× Invalid email or password."

expected to find visible css ".flash .alert" with text "× Invalid email or password." but there were no matches. Also found "× Invalid Email or password.", which matched the selector but not all filters.

# ./spec/features/admin\_spec.rb:49:in `block (2 levels) in <top (required)>'

In a recent version of Devise, the message “Invalid email or password” changed to “Invalid Email or password”. Although not ideal, it is easiest to just fix the tests to allow this slight change to pass.

This was fixed in:

./spec/features/admin\_spec.rb

./spec/features/user\_spec.rb

**#6**

When testing the browser interactions:

1) tracker block should create a new tracker item

Failure/Error: expect(items.length).to be > 0

expected: > 0

got: 0

# ./spec/features/tracker\_block\_spec.rb:99:in `block (2 levels) in <top (required)>'

A recent version of the web browser driver for automated testing appears to have changed its behavior, no longer accepting tab characters in the same way as pressing a tab key on a keyboard. This was resolved in:

./spec/features/tracker\_block\_spec.rb

We now explicitly send a tab key to the input box (and don’t use the **\t** tab character) at lines 87 and 226

fill\_in "master\_player\_infos\_attributes\_0\_last\_name", with: "#{@full\_player\_info.last\_name}"

find("\#master\_player\_infos\_attributes\_0\_last\_name").send\_keys :tab

Note the removal of ‘**\t**’ on the first line.

**#7**

1) tracker block should create a new tracker item

Failure/Error: expect(page).to have\_css 'tbody.index-created[data-template="tracker-result-template"][data-tracker-protocol="'+protocol.name.downcase+'"] .tracker-event\_date', text: "2/2/2030"

expected to find visible css "tbody.index-created[data-template=\"tracker-result-template\"][data-tracker-protocol=\"study\"] .tracker-event\_date" with text "2/2/2030" but there were no matches. Also found "04/08/2017", which matched the selector but not all filters.

# ./spec/features/tracker\_block\_spec.rb:185:in `block (2 levels) in <top (required)>'

This error is due to the user being unable to enter a date through key entries. The ‘**/**’ character is being rejected, which sounds like a familiar bug fixed in the past. This is a genuine error that has been detected.

Start by looking at the previous fix for this bug. It appears that the bug was only resolved for the locale en-US, not en-GB, which this development machine has set for the browser.

Resolved this by fixing (and commenting better) in

public/webshims/shims/combos

Changing

dateSigns: '-'

to

dateSigns: '/'

**#Finally**

...

Finished in 2 minutes 14.1 seconds (files took 2.26 seconds to load)

**525 examples, 0 failures**

Coverage report generated for RSpec to /home/phil/NetBeansProjects/fpa-phase3/coverage. 1993 / 2773 LOC (71.87%) covered.

## Enhancing Test Suite to Improve Browser Tests

In order to provide coverage of some of the components that will be upgraded as Javascript components, a new rspec feature spec has been produced: **spec/features/player\_data\_entry\_spec.rb**

This provides coverage of functionality not previously captured in the UI tests:

* Create MSID from menu
* Edit new Player Info
* Add Address with Canada country, and appropriate region
* datepicker: Edit player and enter birth date
* Chosen (flag picker): Edit player info tags
* Mask (phone number field formatting): Add contact details with phone number
* Typeahead: select a college by typing the first few characters

Table functionality did not have automated test in place. Some basic tests were created before the upgrade to validate success at **reports\_spec.rb**. This tests:

* Report generation into a table
* Sorting ascending and descending on columns

## Removing deprecated Webshims library

The Webshims library is only used for date picker functionality on date input fields. The lack of support and restriction to older versions of jQuery make its replacement a priority.

After some research, the [bootstrap-datepicker](https://github.com/uxsolutions/bootstrap-datepicker) was selected, given its current support and similar appearance and operation to the existing component.

The new library is considerably more compact, and has been placed in **app/assets** alongside other Javascript and CSS. A few small changes were required to **\_fpa\_form\_utils.js** to setup datepickers, and to the main layout to remove the existing Webshims initialization.

Additionally, the file **jquery-swap.js**, containing a patch required for Webshims to work, could also be removed.

## jQuery

In order to upgrade to the most recent version of jQuery (v3.2.1) the following changes are required:

* change app/assets/application.js to reference jquery3
* refactor $.ajax.sucess() calls to use new event callback syntax
* improve modal dialog box and logout dropdown handling to ensure the buttons are fully visible before clicking them in automated tests

## Bootstrap

Bootstrap was configured with custom settings for Zeus. In order to upgrade, the file **vendor/assets/config.json** was uploaded to the Bootstrap customizer at <http://getbootstrap.com/customize/> which provides a download of all the required assets. These replaced the appropriate files in **vendor/assets/**

* javascripts
* css
* fonts

## Typeahead

No upgrade required

## jQuery Chosen

Upgrading to v1.7.0 simply requires download of the new files, then replacing them in **vendor/assets/**

* javascripts/chosen.jquery.js
* css/chosen.css
* images/chosen-sprite.png
* images/chosen-sprite@2x.png

## jQuery Tablesorter

The new version of **jquery.tablesorter.js** (v2.28.15) was pulled from <https://github.com/Mottie/tablesorter/tree/master/js>

After upgrading, the new automated tests were updated and **search\_results.scss** were updated to match the use of new style classes.

## jQuery Scrollto

Pulled version 2.1.2 from <https://github.com/flesler/jquery.scrollTo> and updated the corresponding file in **vendor/assets/javascripts/**

## jQuery Mask

Pulled 1.14.11 from <https://github.com/igorescobar/jQuery-Mask-Plugin> and updated **jquery-mask.js**

Also had to change the features test to send key presses individually, rather than in a chunk, in order to allow the cursor positioning code in the the new version to fire correctly.

## Handlebars

Pulled v4.0.10 from <http://handlebarsjs.com/installation.html> and added it to **vendor/assets/javascripts**. Removed the previous version.

As expected, some breaking changes in the upgrade caused some upgrade challenges. The actual changes required were limited, although took some effort to identify and fix:

* common\_templates/search\_results\_template.html.erb
  + Remove **{{#with ..}}** blocks to respond to breaking change in v4.0
  + Likely due to: <https://github.com/wycats/handlebars.js/issues/1028>
* app/assets/\_fpa\_utils.js
  + Change **\_fpa.utils.pretty\_print** to handle identification of strings, objects and numbers better
  + Likely due to: <https://github.com/wycats/handlebars.js/issues/1093>

# Running Tests

## Standard Tests

The full set of rspec tests (including the new *features* tests) was run after each upgrade, and after all were completed. A complete end to end run without errors was successful.

## Extended Tests Checking Warnings

To attempt to identify code issues that may be hiding, running tests that display Ruby interpreter *warnings* can provide extra information.

The tests were re-run with the configuration:

**config.warnings = true**

This led to many warnings being fired. The vast majority were related to uninitialized instance variables being used. Although this is a very common use (most developers just assume that **@some\_variable** automatically initializes to nil the first time it is touched), in order to truly use the Ruby warnings it was necessary to explicitly initialize the variables when models and controllers were created.

The file **config/initializers/app\_var\_initialization.rb** re-opens **class Object** to add a method that allows initialization of instance variables is they are not already defined.

class Object

def instance\_var\_init var, o=nil

varname = "@#{var}"

unless instance\_variable\_defined?(varname)

instance\_variable\_set(varname, o)

end

end

end