PostgreSQL FTS Relay Application

PostgreSQL FTS Relay web application is built on Spring MVC framework module. The application gets compiled on JDK 1.8 through Maven, and runs on any environment where the JDK is available. The only external dependency for the application – is external PostgreSQL database.

Configuration

Application has only single configuration file – *application.properties,* where postgre credentials are supplied to the application. When running application on various environment it is suggested to initialize postgre settings not via properties, but through the environment variables.

When compiled, the application may run via command:

> *java -D**database.user**=tnkxkfefnfysiw -D**database.pwd**=eoyUq2ZLcYcPY7U0bvxtRhLAxx -D**database.url**=jdbc:postgresql://ec2-54-225-255-208.compute-1.amazonaws.com:5432/d85lilb3bg1tg0 $JAVA_OPTS -jar target/dependency/jetty-runner.jar --port $PORT target/*.war*

Source Repository

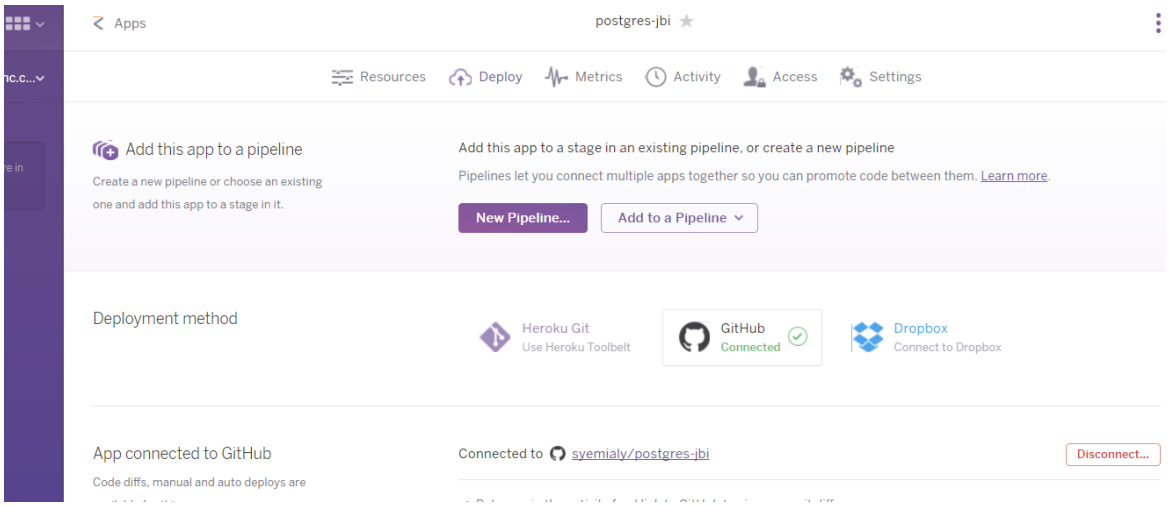
The sources for the application are in GitHub https://github.com/syemialy/postgres-jbi To work with the project you may fork the repository or request private access to it.

Deployment Instructions

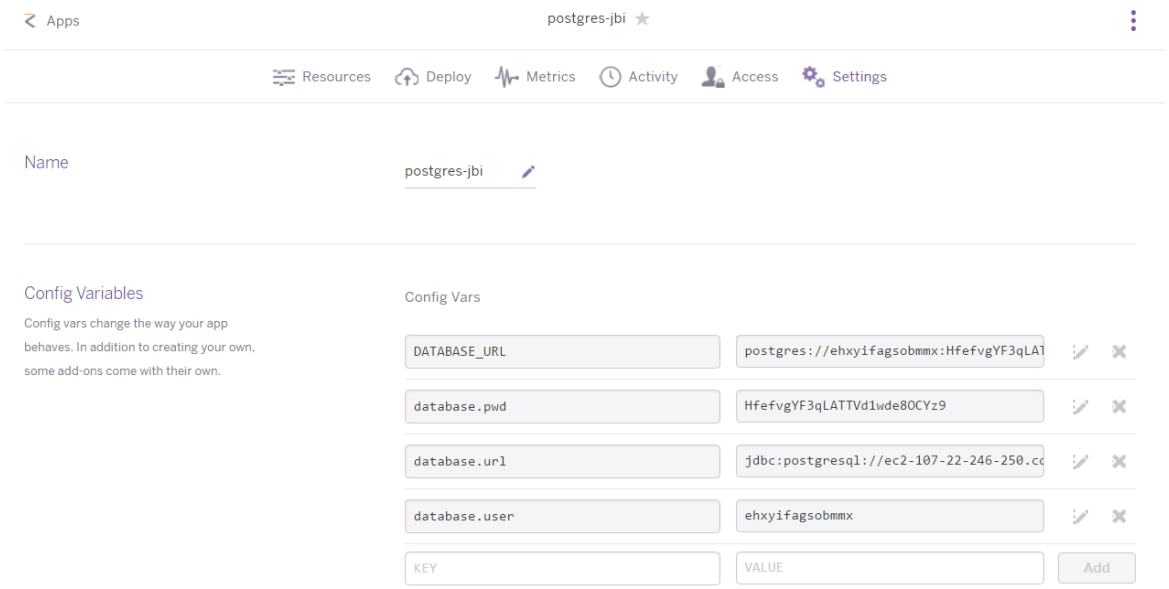
Deployment to Heroku via GitHub

The deployment instructions assume you have forked the ripository **postgres-jbi**

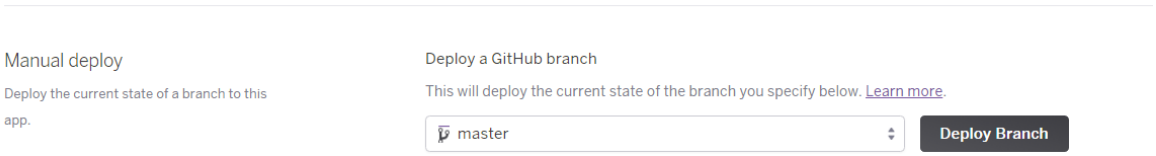
- Create new Heroku application
- On the application settings page, navigate to Deploy section
    - Link your GitHub repository



- On the application Settings section, add all three environment variables .



- Go Back to Deploy section and execute manual deploy of your new application from your github ripository



Once your application is deployed, you may use curl to validate all the REST services. Samples of the curl usage may be found in docs/curl.samples.txt file.

REST Service Endpoints

Endpoint	Method	Format	Description
srv/search	POST	JSON	A text query passed in JSON body is used to create FTS search request against underlying PostgreSQL database
srv/index	POST	JSON	Create index on documents, created based on the selected columns. Columns are added to the ts_vector with coalesce() function
srv/index	DELETE	JSON	Drops the index by name

Data Format for JSON messages

Search Request

Field			Type	Mandatory	Description
table			Object	Yes	
	name		String	Yes	Name of the table against which the FTS search will be executed
	columns		Object[]	Yes	Column name(s) to create a document (unit of searching)
		name	String	Yes	Name of the column within the table
		selectable	Boolean	Yes	Indicates if column will be in SELECT statement predicate
		tsvectorinclude	Boolean	Yes	Indicates if column will be used to create ts_vector
query			String	Yes	User-written text to be used in *plain_tsquery* function to create search terms
configuration			String	No	Configuration name to be used to parse and normalize strings. By default it will be set to *english*

```
{ “table”: {
        “name”: “products”,
        columns:[
                {“name”:”product_name”, “selectable”:true, “tsvectorinclude”:true},
                {“name”:”description”, “selectable”:false, “tsvectorinclude”:true},
                {“name”:”product_id”, “selectable”:true, “tsvectorinclude”:false}],

        “query”:”sold by pair”,
        “configuration”:”english”
        }
```

Search Response

Field		Type	Mandatory	Description
result		Object	Yes	
	sqlstatement	String	Yes	Name of the table against which the FTS search will be executed
	records	Object[]	Yes	
	javatimemls	Integer	Yes	Indicates time the java program ran FTS search, this is to collect statement statistics
error		Boolean	Yes	User-written text to be used in *plain_tsquery* function to create search terms
error_message		String	No	Configuration name to be used to parse and normalize strings. By default it will be set to *english*

Create Index Request

Field	Type	Mandatory	Description
name	String	Yes	Name of the index to be created
type	String	No	Type of the index to be created, GIN will be default type.
table	Object	Yes	Name of the table for which index will be created
name	String	Yes	Name of the table against which the FTS search will be executed
columns	Object[]	Yes	Column name(s) to create a document (unit of searching)
name	String	Yes	
configuration	String	No	Configuration name to be used to parse and normalize strings. By default it will be set to *english*

```
{"name":"idx_prddescr",
 "table":{"name":"products",
          "columns":[
                    {"name":"description"},
                     {"name":"product_name"}
]}}
```

Create Index Response

Field	Type	Mandatory	Description
result	String	Yes	Status of the request
sqlstatement	String	Yes	The SQL statement which was executed
error	Boolean	Yes	User-written text to be used in *plain_tsquery* function to create search terms
error_message	String	No	Configuration name to be used to parse and normalize strings. By default it will be set to *english*

```
{"result":"created",
 "error":false,
 "sqlstatement":"CREATE INDEX idx_prddescr ON products USING GIN
(to_tsvector(\u0027english\u0027,coalesce(description,\u0027\u0027) || \u0027 \u0027 ||
coalesce(product_name,\u0027\u0027)))"
}
```

Drop Index Request

Field	Type	Mandatory	Description
name	String	Yes	Name of the index to be dropped

{"name":"idx_prddescr"}

Drop Index Response

Field	Type	Mandatory	Description
result	String	Yes	Status of the request
sqlstatement	String	Yes	The SQL statement which was executed
error	Boolean	Yes	User-written text to be used in *plain_tsquery* function to create search terms
error_message	String	No	Configuration name to be used to parse and normalize strings. By default it will be set to *english*

{"result":"dropped",
"error":false,
"sqlstatement":"DROP INDEX idx_prddescr"}