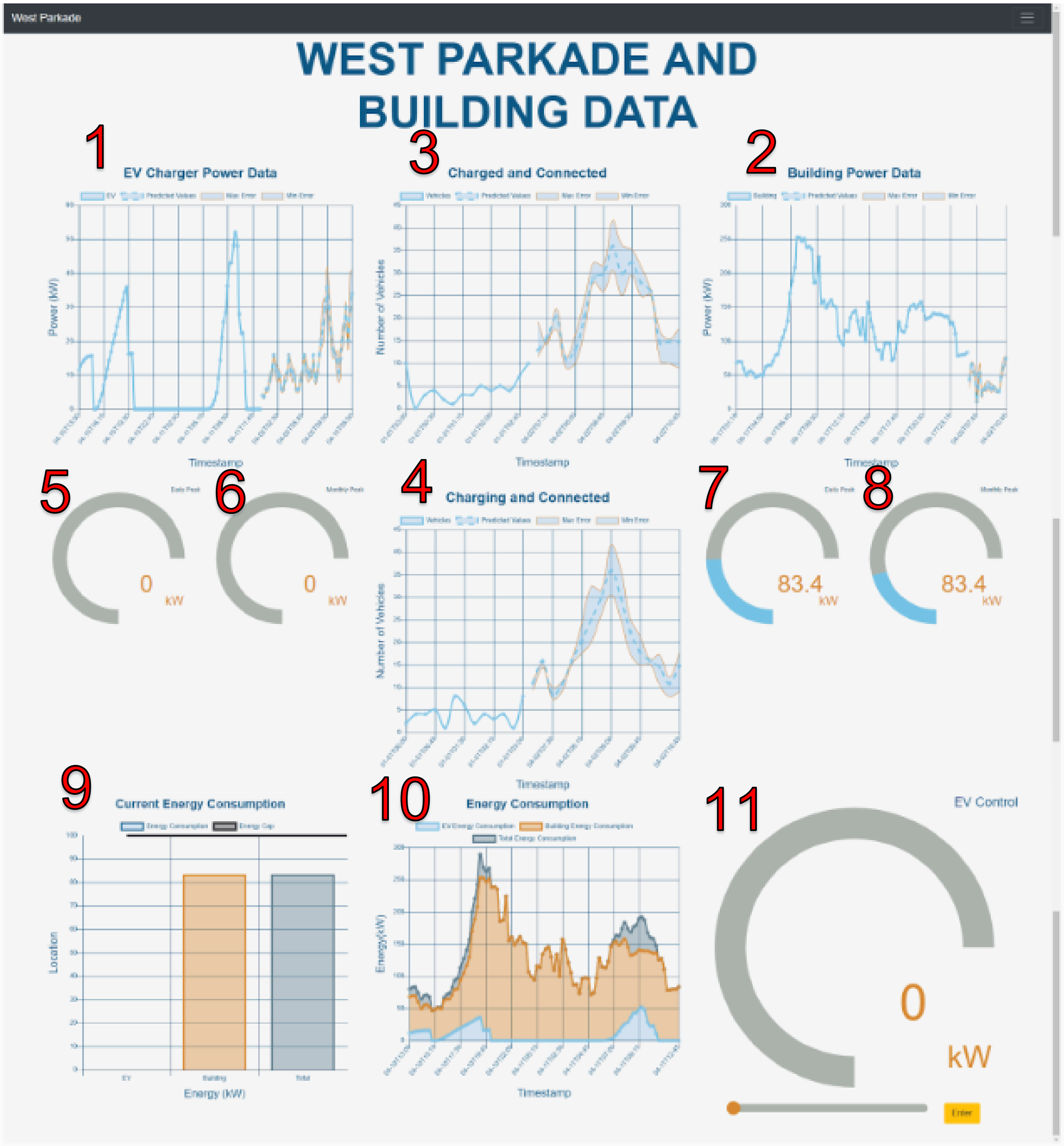
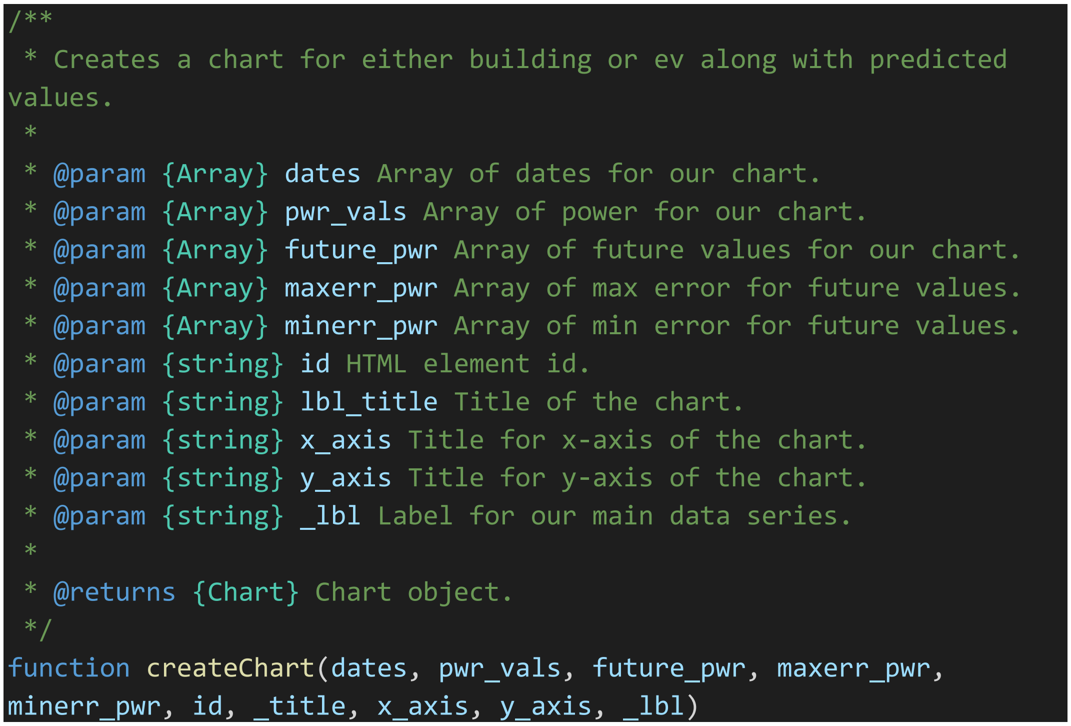


In this document, there will be a description and explanation for each of the 11 components on the page, as well as at the end, what each of these components are linked to for our 5 parkades.

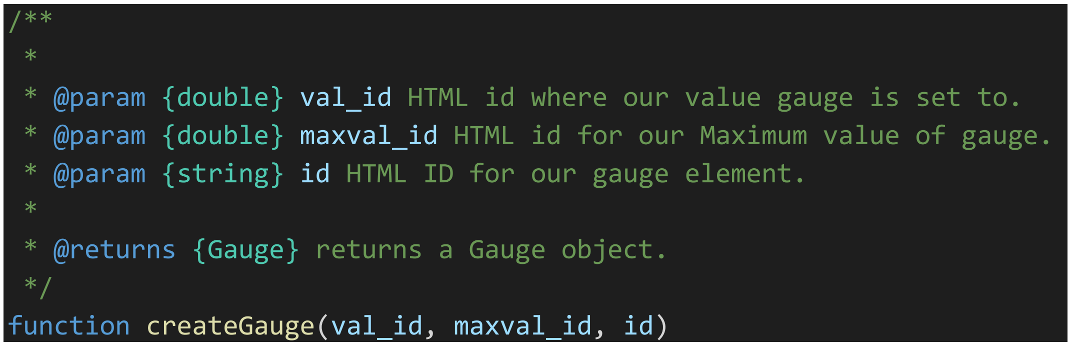
1. Realtime EV data
   - To modify where to GET the real time/historic data, in the HTML template {% block jquery %}, change the **var evpoint = {new url}**
   - To modify where to GET the predicted data, in the HTML template {% block jquery %}, change the **var evpredictedpoint = {new url}**
   - To edit the charts labels, in the HTML template {% block jquery %}, edit the last four variables in **var evChart = createChart(*args, label title, x-axis label, y-axis label, label for the main data series)**



```
/**
 * Creates a chart for either building or ev along with predicted values.
 *
 * @param {Array} dates Array of dates for our chart.
 * @param {Array} pwr_vals Array of power for our chart.
 * @param {Array} future_pwr Array of future values for our chart.
 * @param {Array} maxerr_pwr Array of max error for future values.
 * @param {Array} minerr_pwr Array of min error for future values.
 * @param {string} id HTML element id.
 * @param {string} lbl_title Title of the chart.
 * @param {string} x_axis Title for x-axis of the chart.
 * @param {string} y_axis Title for y-axis of the chart.
 * @param {string} _lbl Label for our main data series.
 *
 * @returns {Chart} Chart object.
 */
function createChart(dates, pwr_vals, future_pwr, maxerr_pwr, minerr_pwr, id, _title, x_axis, y_axis, _lbl)
```

2. Realtime Building data
   - Similar to 1, except modify the following variables:
   - **var buildingpoint**
   - **var bdpredictedpoint**
   - **var bdChart = createChart()**
3. Realtime Charged (finished) and Connected Vehicle data
   - Similar to 1, except modify the following variables:
   - **var chargedcarpoint**
   - **var chargedpredictedpoint**
   - **var chargedcarChart = createChart()**
4. Realtime Charging and Connected Vehicle data
   - Similar to 1, except modify the following variables:
   - **var chargingcarpoint**
   - **var chargingpredictedpoint**
   - **var chargingcarChart = createChart()**
5. Realtime Daily EV gauge
   - To modify our gauge, in the HTML template {% block content %} there is a element with **<div id="evdailygauge" …. > …. </div>**

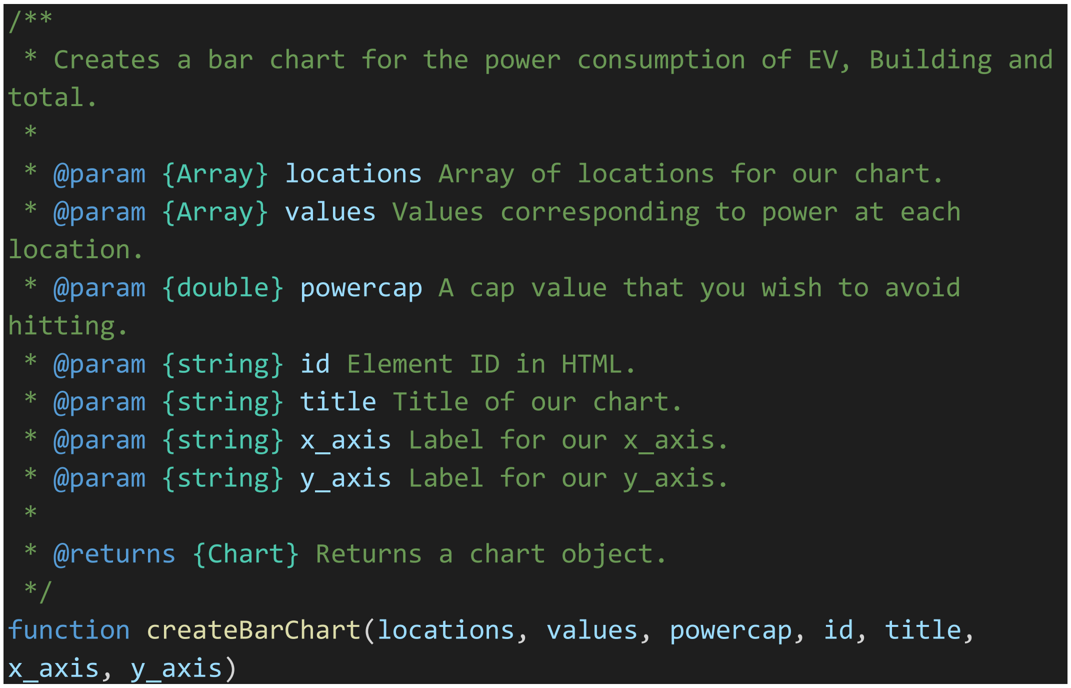
- To modify our labels
    - To change the units (ie. kW to kWh) modify the **<span class="value-text">units here</span>**
    - To change the label modify the **<span class="label">label here</span>**
- To change the current value, modify the **<input type="hidden" id="evdailyval" value={{curr_ev}}>**, note that the **{{curr_ev}}** is what's being sent from *views.graph.getgraph()* by that name.
- To change the maximum value, modify the **<input type="hidden" id="evdailymax" value={{max_evdaily}}>**, similarly note the **{{max_evdaily}}** is the name sent from *views.graph.getgraph()*.
- Note: If you change the **id** of any of the elements, you will need to find that element id in the rest of the template and change it accordingly as well, particularly in the **createGauge()** in the {% block jquery %}



```
/**
 *
 * @param {double} val_id HTML id where our value gauge is set to.
 * @param {double} maxval_id HTML id for our Maximum value of gauge.
 * @param {string} id HTML ID for our gauge element.
 *
 * @returns {Gauge} returns a Gauge object.
 */
function createGauge(val_id, maxval_id, id)
```

6. Realtime Monthly EV gauge
    - Similar to 5, but with the following variables:
    - **<div id="evmonthlygauge" class="gauge-container">**
    - **<input type="hidden" id="evmonthlyval" value={{curr_ev}}>**
    - **<input type="hidden" id="evmonthlymax" value={{max_evmonthly}}>**
    - Labels are the same just in the corresponding element
7. Realtime Daily Building gauge
    - Similar to 5, but with the following variables:
    - **<div id="bddailygauge" class="gauge-container">**
    - **<input type="hidden" id="bddailyval" value={{curr_bd}}>**
    - **<input type="hidden" id="bddailymax" value={{max_bddaily}}>**
    - Labels are the same just in the corresponding element
8. Realtime Monthly Building gauge
    - Similar to 5, but with the following variables:
    - **<div id="bdmonthlygauge" class="gauge-container">**
    - **<input type="hidden" id="bdmonthlyval" value={{curr_bd}}>**
    - **<input type="hidden" id="bdmonthlymax" value={{max_bdmonthly}}>**
    - Labels are the same just in the corresponding element
9. Realtime Bar chart
    - To modify the bar chart, in the HTML template {% block jquery %} the **var barChart = createBarChart()**

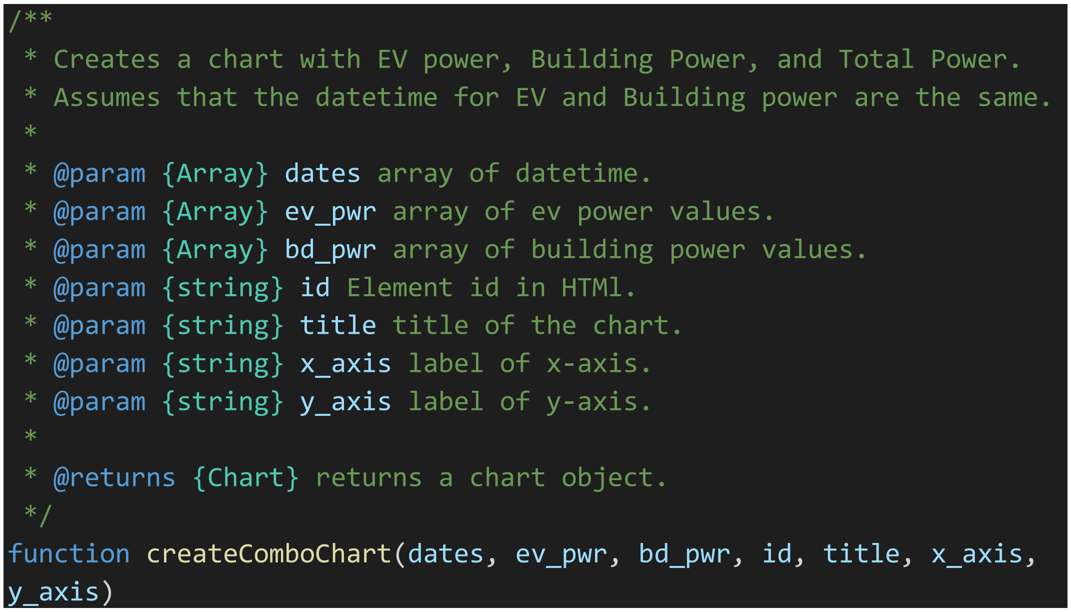
- Most importantly is the parameter *powercap*, which is what the line will be set to



```
/**
 * Creates a bar chart for the power consumption of EV, Building and
total.
 *
 * @param {Array} locations Array of locations for our chart.
 * @param {Array} values Values corresponding to power at each
location.
 * @param {double} powercap A cap value that you wish to avoid
hitting.
 * @param {string} id Element ID in HTML.
 * @param {string} title Title of our chart.
 * @param {string} x_axis Label for our x_axis.
 * @param {string} y_axis Label for our y_axis.
 *
 * @returns {Chart} Returns a chart object.
 */
function createBarChart(locations, values, powercap, id, title,
x_axis, y_axis)
```

10. Realtime Combination chart

- To modify the combination chart, in the HTML template {% block jquery %} the **var testChart = createComboChart()**



```
/**
 * Creates a chart with EV power, Building Power, and Total Power.
 * Assumes that the datetime for EV and Building power are the same.
 *
 * @param {Array} dates array of datetime.
 * @param {Array} ev_pwr array of ev power values.
 * @param {Array} bd_pwr array of building power values.
 * @param {string} id Element id in HTMl.
 * @param {string} title title of the chart.
 * @param {string} x_axis label of x-axis.
 * @param {string} y_axis label of y-axis.
 *
 * @returns {Chart} returns a chart object.
 */
function createComboChart(dates, ev_pwr, bd_pwr, id, title, x_axis,
y_axis)
```

11. Control gauge

- Similar to 5, but with the following variables for the gauge
- **<div id="evcontrolgauge" class="gauge-container">**
- Labels:

    - **<span class="large-value-text">units here</span>**
    - **<span class="large-label">label here</span>**
- **<input type="hidden" id="evcurrval" value={{curr_ev}}>**
- **<input type="hidden" id="evmaxval" value="200">**, change this according to however you want to set the max value into
- The slider variables are found just below the control gauge element in HTML
- **<input type="range" min="0" max="200" step="0.001" value={{curr_ev}} class="slider" id="TestEV">** is the slider
- **<button type="button" class="btn btn-warning">Enter</button>** corresponds to the button
- In the {% block jquery %} near the end, there is a function that controls the gauge and slider interaction, **evslider.oninput = function () { evctrlgauge.setValue(this.value) }**

Parkades (Pages)

Note that for the ***path*** function, the variables work as follows:

1. **r'…'** : This is the URL path that the view will direct to, once the app is running you can look at this view by navigating to this url
2. **views…** : This is the function in views.py that will return the data and also possibly map the url to the a template of some sorts
3. **name**: this is how you can easily refer to the url from both front and backend in order to query data (in our case, we use it in our html templates to jquery realtime data)

West Parkade (/west):

1. Current/Past 24 hours EV data is linked to
   ```
   path(r'api/west', views.EVData.west, name='west'),
   ```
   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/west', views.EVPredictedData.west,
   name='west_future'),
   ```

2. Current/Past 24 hours Building data is linked to
   ```
   path(r'api/buildingwest', views.BuildingData.west,
   name='buildingwest'),
   ```
   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/buildingwest', views.BuildingPredictedData.west,
   name='buildingwest_future'),
   ```
3. Current/Past 24 hours Charged and Connected vehicles data is linked to
   ```
   path(r'api/chargedcars/west',views.chargedCarsData.west,
   name='carschargedwest'),
   ```
   Predicted 12 hours data is linked to

```
path(r'api/prediction/chargedcars/west',
views.chargedCarsPredictedData.west, name='carschargedwest_future'),
```

4. Current/Past 24 hours Charging and Connected vehicles data is linked to

   ```
   path(r'api/chargingcars/west',views.chargingCarsData.west,
   name='carschargingwest'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/chargingcars/west',
   views.chargingCarsPredictedData.west, name='carschargingwest_future'),
   ```

5. EV Current to Daily Peak Gauge is linked to **graphs.urls.graph.getgraph(***request, parkade, template***)**, specifically the 'curr_ev': ev_pwr and 'max_evdaily': ev_daily
6. EV Current to Monthly Peak Gauge is linked to 'curr_ev': ev_pwr and 'max_evmonthly': ev_monthly
7. Building Current to Daily Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bddaily': bd_daily
8. Building Current to Monthly Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bdmonthly': bd_monthly
9. Realtime Bar Chart is linked to the most recent value from api/west and api/buildingwest
10. Realtime Combination Chart is linked to the data from api/west and api/buildingwest
11. Control Gauge is linked to 'curr_ev': ev_pwr

Rose Parkade (/rose):

1. Current/Past 24 hours EV data is linked to

   ```
   path(r'api/rose', views.EVData.rose, name='rose'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/rose', views.EVPredictedData.rose,
   name='rose_future'),
   ```

2. Current/Past 24 hours Building data is linked to

   ```
   path(r'api/buildingrose', views.BuildingData.rose,
   name='buildingrose'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/buildingrose', views.BuildingPredictedData.rose,
   name='buildingrose_future'),
   ```

3. Current/Past 24 hours Charged and Connected vehicles data is linked to

   ```
   path(r'api/chargedcars/rose',views.chargedCarsData.rose,
   name='carschargedrose'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/chargedcars/rose',
   views.chargedCarsPredictedData.rose, name='carschargedrose_future'),
   ```

4. Current/Past 24 hours Charging and Connected vehicles data is linked to

```
path(r'api/chargingcars/rose',views.chargingCarsData.rose,
name='carschargingrose'),
```

Predicted 12 hours data is linked to

```
path(r'api/prediction/chargingcars/rose',
views.chargingCarsPredictedData.rose, name='carschargingrose_future'),
```

5. EV Current to Daily Peak Gauge is linked to **graphs.urls.graph.getgraph(***request, parkade, template***)**, specifically the 'curr_ev': ev_pwr and 'max_evdaily': ev_daily
6. EV Current to Monthly Peak Gauge is linked to 'curr_ev': ev_pwr and 'max_evmonthly': ev_monthly
7. Building Current to Daily Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bddaily': bd_daily
8. Building Current to Monthly Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bdmonthly': bd_monthly
9. Realtime Bar Chart is linked to the most recent value from api/rose and api/buildingrose
10. Realtime Combination Chart is linked to the data from api/rose and api/buildingrose
11. Control Gauge is linked to 'curr_ev': ev_pwr

Health Parkade (/health):

1. Current/Past 24 hours EV data is linked to

   ```
   path(r'api/health', views.EVData.health, name='health'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/health', views.EVPredictedData.health,
   name='health_future'),
   ```

2. Current/Past 24 hours Building data is linked to

   ```
   path(r'api/buildinghealth', views.BuildingData.health,
   name='buildinghealth'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/buildinghealth',
   views.BuildingPredictedData.health, name='buildinghealth_future'),
   ```

3. Current/Past 24 hours Charged and Connected vehicles data is linked to

   ```
   path(r'api/chargedcars/health',views.chargedCarsData.health,
   name='carschargedhealth'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/chargedcars/health',
   views.chargedCarsPredictedData.health,
   name='carschargedhealth_future'),
   ```

4. Current/Past 24 hours Charging and Connected vehicles data is linked to

   ```
   path(r'api/chargingcars/health',views.chargingCarsData.health,
   name='carscharginghealth'),
   ```

   Predicted 12 hours data is linked to

```
path(r'api/prediction/chargingcars/health',
views.chargingCarsPredictedData.health,
name='carschargninghealth_future'),
```

5. EV Current to Daily Peak Gauge is linked to **graphs.urls.graph.getgraph(***request, parkade, template***)**, specifically the 'curr_ev': ev_pwr and 'max_evdaily': ev_daily
6. EV Current to Monthly Peak Gauge is linked to 'curr_ev': ev_pwr and 'max_evmonthly': ev_monthly
7. Building Current to Daily Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bddaily': bd_daily
8. Building Current to Monthly Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bdmonthly': bd_monthly
9. Realtime Bar Chart is linked to the most recent value from api/health and api/buildinghealth
10. Realtime Combination Chart is linked to the data from api/health and api/buildinghealth
11. Control Gauge is linked to 'curr_ev': ev_pwr

Fraser Parkade (/fraser):

1. Current/Past 24 hours EV data is linked to
   ```
   path(r'api/fraser', views.EVData.fraser, name='fraser'),
   ```
   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/fraser', views.EVPredictedData.fraser,
   name='fraser_future'),
   ```

2. Current/Past 24 hours Building data is linked to
   ```
   path(r'api/buildingfraser', views.BuildingData.fraser,
   name='buildingfraser'),
   ```
   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/buildingfraser',
   views.BuildingPredictedData.fraser, name='buildingfraser_future'),
   ```
3. Current/Past 24 hours Charged and Connected vehicles data is linked to
   ```
   path(r'api/chargedcars/fraser',views.chargedCarsData.fraser,
   name='carschargedfraser'),
   ```
   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/chargedcars/fraser',
   views.chargedCarsPredictedData.fraser,
   name='carschargedfraser_future'),
   ```
4. Current/Past 24 hours Charging and Connected vehicles data is linked to

   ```
   path(r'api/chargingcars/fraser',views.chargingCarsData.fraser,
   name='carschargingfraser'),
   ```
   Predicted 12 hours data is linked to

```
path(r'api/prediction/chargingcars/fraser',
views.chargingCarsPredictedData.fraser,
name='carschargingfraser_future'),
```

5. EV Current to Daily Peak Gauge is linked to **graphs.urls.graph.getgraph(***request, parkade, template***)**, specifically the 'curr_ev': ev_pwr and 'max_evdaily': ev_daily
6. EV Current to Monthly Peak Gauge is linked to 'curr_ev': ev_pwr and 'max_evmonthly': ev_monthly
7. Building Current to Daily Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bddaily': bd_daily
8. Building Current to Monthly Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bdmonthly': bd_monthly
9. Realtime Bar Chart is linked to the most recent value from api/fraser and api/buildingfraser
10. Realtime Combination Chart is linked to the data from api/fraser and api/buildingfraser
11. Control Gauge is linked to 'curr_ev': ev_pwr

North Parkade (/north):

1. Current/Past 24 hours EV data is linked to

   ```
   path(r'api/north', views.EVData.north, name='north'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/north', views.EVPredictedData.north,
   name='north_future'),
   ```

2. Current/Past 24 hours Building data is linked to

   ```
   path(r'api/buildingnorth', views.BuildingData.north,
   name='buildingnorth'),
   ```

   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/buildingnorth',
   views.BuildingPredictedData.north, name='buildingnorth_future'),
   ```

3. Current/Past 24 hours Charged and Connected vehicles data is linked to

   ```
   path(r'api/chargedcars/north',views.chargedCarsData.north,
   name='carschargednorth'),
   ```

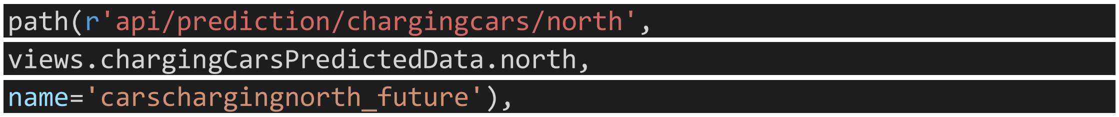
   Predicted 12 hours data is linked to

   ```
   path(r'api/prediction/chargedcars/north',
   views.chargedCarsPredictedData.north, name='carschargednorth_future'),
   ```

4. Current/Past 24 hours Charging and Connected vehicles data is linked to

   ```
   path(r'api/chargingcars/north',views.chargingCarsData.north,
   name='carschargingnorth'),
   ```

   Predicted 12 hours data is linked to



```
path(r'api/prediction/chargingcars/north',
views.chargingCarsPredictedData.north,
name='carschargingnorth_future'),
```

5. EV Current to Daily Peak Gauge is linked to **graphs.urls.graph.getgraph(***request, parkade, template***)**, specifically the 'curr_ev': ev_pwr and 'max_evdaily': ev_daily
6. EV Current to Monthly Peak Gauge is linked to 'curr_ev': ev_pwr and 'max_evmonthly': ev_monthly
7. Building Current to Daily Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bddaily': bd_daily
8. Building Current to Monthly Peak Gauge is linked to 'curr_bd': bd_pwr and 'max_bdmonthly': bd_monthly
9. Realtime Bar Chart is linked to the most recent value from api/north and api/buildingnorth
10. Realtime Combination Chart is linked to the data from api/north and api/buildingnorth
11. Control Gauge is linked to 'curr_ev': ev_pwr