DATA INPUT

Open `input_data.json` to start editing your input

1. Starting point

Point at which the detailing starts within the model space. Details are drawn around the starting point. By default starting is `0,0,0` for `x, y` and `z` coordinates

2. Sections data

All sections data is entered under `sections` section in the `input_data.json` file.

Section Input parameters

- **b** stands for width and "**d**" stands for depth
- **f**" stands for flange and "**w**" stands for web
- **b** - total width of section //may not be applicable for end user
- **bf_top** - width of flange on top
- **bf_bottom** - width of flange at the bottom
- **bw** - width of web
- **d** - total depth of section
- **df** - depth of flange
- **w_offset** - off set of web from left starting point of section

NOTE: `bf_top`, `bf_bottom` and `bf` will or may mean the same thing in this document

Specification for square/rectangular sections

When no `df` is specified, section is assumed to be square or rectangular

```
 '''
             bw
       2  -------  3
         |       |
         |       |
section  |       |
depth(d) |       |
         |       |
      1   -------  4
 '''
```

- example input for square section. See `section_4` in `sample_file.json`

```
        "section_x": {
            "bw": "200",
            "d": "450"
        }
```

Specification for L Right Flange Sections

L Right secctions have no web offset (`w_offset`) and `bw` is smaller than `bf`

```
        '''
                   bf
          2  ------------------- 3
            |                   | df
            |                   |
            |        ----------- 4
   section  |       | 5
   depth(d) |       |
            |       |
            |       |
            |       |
         1   -------  6
               bw
        '''
```

- example input for L right flanged section. See `section_2` in `sample_file.json`

```
        "section_x": {
            "bf_top": "350",
            "bw": "200",
            "d": "450",
            "df": "250"
        }
```

Specification for L Left Flange Sections

L Left secctions' `w_offset` and `bw` total up to `bf`.

```
        '''

        Vertices for L section left flange
                  bf
          4  -------------------   5
         df |                   |
            |                   |
  3  -----------        |
   w_offset   2 |       |  section
                |       |  depth(d)
                |       |
                |       |
                |       |
              1  -------   6
                   bw
'''
```

- example input for L right flanged section. See `section_3` in `sample_file.json`
- `bw + w_offset = 350` and `bf_top = 350` meaning it's an L right section

```
"section_x": {
    "bf_top": "350",
    "bw": "200",
    "d": "450",
    "df": "200",
    "w_offset": "150"
}
```

Specification for T sections

if it is not a square or L section, then it is a T section. T sections `w_offset` and `bw` do not total upto `bf`

```
'''

Vertices for T sections
                bf
  4  -------------------------------    5     |
    |                               | df      |
    |                               |         |
  3  -----------         -----------   6      |
    w_offset  2 |       |  7                  |
                |       |                     |  section
                |       |                     |  depth(d)
                |       |                     |
                |       |                     |
              1  -------   8
                   bw
'''
```

- example input for T flanged section. See `section_1` in `sample_file.json`

```
"section_x": {
    "bf_top": "500",
    "bw": "200",
    "d": "450",
    "df": "200",
    "w_offset": "150"
}
```

NOTE: It is possible to enter values as negative or positive. Positive x is towards the right and positive y is towards the top.

3. Supports types data

Support types specify the column properties at each support point. The support point could have a column at the bottom and no column on top. The Column at the bottom could have a wider section than the column on top.

Support types specify these properties including the section properties of the column. At the this point in time, there is only consideration for *rectangular/square* column sections. Circular sections will be added as need arises.

Column Input parameters

- **column_top**: Specifications for column at the top of support. If no value this parameter is not provided, then the support type has no column on bottom.
- **column_bottom**: Specifications for column at the bottom of support. If no value this parameter is not provided, then the support type has no column on top.
- **section_d**: This stands for the section depth of a column
- **section_b**: This stands for section width of column
- **column_h_m**: This is the height of the column in **metres**

NOTE: Only column section width (*section_b*) is being used in beam detailing the time of writing this text. Other parameters will be used when column detailing has been added to the "*adds*" project.

- example input for support type with both column on top and the bottom. See `support_type_1` in `sample_file.json`

```
        "support_type_x":{
            "column_top": {
                "section_d": "200",
                "section_b": "200",
                "column_h_m": "3"
            },
            "column_bottom": {
                "section_d": "200",
                "section_b": "200",
                "column_h_m": "3"
            }
        }
```

- example input for support type with only the bottom bottom column. See `support_type_2` in `sample_file.json`

```
        "support_type_x":{
            "column_bottom": {
                "section_d": "200",
                "section_b": "300",
                "column_h_m": "3"
            }
        }
```

4. Beams data

Beams sections is where the beams are specified using both the **sections data** and **support types data** plus other beam parameters required to exaustively specify a beam.

Beam Input parameters

- **beam_depth**: This is the overall depth of beam. Some sections a long length of beam may be shallower.
- **spans**: Specifies all the spans in beam. The have their own parameter as shown below.
    - *length_m*: This is the span *centre* to *centre* length in *metres*. Or this is the distance between the centre of the of left support to the centre of the right support.
    - *section_left*: The is the section to the left of the span as specified in the sections section. The section is specified using the name of the section.
    - *section_right*: The is the section to the right of the span as specified in the sections section. The section is specified using the name of the section.
    - example input for span parameters. See `span_1` of `beam_1` in `sample_file.json`

```
        "span_x":{
                "length_m": "4.15",
                "section_left": "section_2",
                "section_right": "section_1"
        },
```

NOTE: span key words `length_m`, `section_left` and `section_right` can not be changed to other names

- **supports**: This is specification for the support type at each support point of the beam. A beam with with 3 spans will have 4 supports. Specifying less supports will throw an error.

Contact isaac in case of questions