

# JSON-Calculation Scheme

This concept is used in a number of calculation formats in the Fortune-Teller API.

Generally speaking, a JSON-File, that is committed in the GIT project contains a calculation structure, that is to be applied on a base value. The structure provided in the JSON-File provides the raw-data for the calculation scheme, so that it is highly adjustable, without impacting the APIs code.

---

## API calculation engine

The scheme calculator is called with Localization (i.e. DE), with the calculation type, a base value and a number of variables. The input items are checked, the versions are identified and errors / alternatives raised if not available. As a result the calculation is performed and a number of labelled items and the values are given back, along with messages raised, disclaimer and sources.

Step Elements	Possible Values	Example
type	<i>Absolute if range is met, this value or the referenced labelvalue is added Percent a percentage is applied on an amount in the range StepPercent a percentage is applied for each part of the range that is below the base value Add if range is met, the label value is added Multiply if range is met, the label value is added</i>	
base	<i>label id any label available as input or in the calculation</i>	
from, to	<i>Decimal value lower and upper range of applicable base value. Range includes these values.</i>	
part	<i>Decimal value of 1 or below applied on the range of the base amount (only percent or absolute amount)</i>	
label	<i>label id result is cumulatively saved under this label</i>	
var	<i>Decimal value i.e. an amount or a decimal below 1 for a rate to be applied.</i>	

labelvar	label id any label available as input or in the calculation	
----------	---	--

## Core Elements of the JSON file (in folder jsonlib)

1. The name of the file is defined in the respective module, but begins with the Country Code (2-Letter)
2. The Json begins with a context depending on the scope of the settings included, any number of generic items can be listed.
3. The Input section declares expected input variables and whether these are obligatory
4. The actual type of calculations is named and provides a collection of versions
5. Each version includes a collection of steps, that will be followed consecutively
6. Each calculation scheme ends with a Disclaimer and Source tag, this must contain any limitations to the accuracy of the calculation and the source of the information included in the scheme.

### Example for a file (not making a lot of sense)

```
{
  "country": "Deutschland",
  "comment1": "Order versions from recent to old, followed by Disclaimer",
  "input":
  [
    {
      "label": "basevalue",
      "obligatory": "no"
    },
    [
      {
        "label": "prepayment",
        "obligatory": "yes"
      }
    ],
    {
      "income":
      [
        {
          "2021": [
            [
              [
                [
                  "2020": [
                    [
                      {
                        "type": "absolute",
                        "base": "value",
```

```

    "from": "0",
    "to": "1200",
        "part": "1",

    "label": "gez",
    "var": "120"
    },
    {
        "type": "percent",
            "base": "value",

        "from": "0",
        "to": "1200",
            "part": "1",

        "label": "est",
        "var": "0.12"
        },
        {
            "type": "percent",
                "base": "est",

            "from": "0",
            "to": "999999999",
                "part": "1",

            "label": "soli",
            "var": "0.05"
        }
    ],
    "Disclaimer": "Dieser Text erkl\u00e4rt Besonderheiten",
        "Source": "EStG, Dejure.org 31.7.2021"
}
}
}

```

Revision #9

Created 31 July 2021 19:29:54 by Hendrik

Updated 1 August 2021 21:52:11 by Hendrik