Documentation of Electron JSON Files

In this documentation, we will introduce two JSON files uploaded to Electron, namely backtesting file for backtesting page, and Monte Carlo file for Monte Carlo page.

1 Backtesting Page

The backtesting file has two objects: "parameter" and "result".A template file is as followed:

static > {} backtesting.json > {} result


```
{
    "parameter":
    {
        "structureInput": "classic",
        "codeInput": "000905.XSHG",
        "durationInput": 6,
        "start_obInput": 3,
        "div_couponInput": 0.06,
        "ko_couponInput": 0.06,
        "kiInput": 0.8,
        "koInput": 1.03,
        "sdateInput": "01/01/2022",
        "edateInput": "12/31/2022",
        "sdInput": "",
        "ko_coupon2Input": 0.04,
        "coupon2_periodInput": 3,
        "ko_coupon3Input": "",
        "coupon3_periodInput": "",
        "guaranteedInput": "",
        "limitedlossInput": "",
        "parachuteInput": ""
    },
    "result":
    {
        "0": {
            "start_date": 1641254400000,
            "close": 7354.4562,
            "payoff": -0.1130369231,
            "state": "ki_noko",
            "actual_end_date": 1656892800000
        },
        "1": {
            "start_date": 1641340800000,
            "close": 7222.6747,
            "payoff": -0.0984600622,
            "state": "ki_noko",
            "actual_end_date": 1656979200000
        },
        "2": {
            "start_date": 1641427200000,
```


Figure 1: Backtesting file template

1.1 Parameter Object

All of the parameters are permitted to be null if needed, and the renderer process of JavaScript will automatically judge which parameter is useful and whether the inputs are valid according to the structure.

Table 1: Backtesting Parameter JSON

JSON Key	Context	Example
structureInput	(str) structure	"classic"
codeInput	(str) stock code	"000905.XSHG"
durationInput	(int) contract duration	12
start_obInput	(int) month starting observation	3
div_couponInput	(float) dividend coupon	0.06
ko_couponInput	(float) knock-out coupon	0.06
kiInput	(str or float) knock-in price	"" or 0.8
koInput	(float) knock-out price	1.03
sdateInput	(str) start date: MM/DD/YYYY	"01/01/2022"
edateInput	(str) end date: MM/DD/YYYY	"12/31/2022"
sdInput	(str or float) stepdown	"" or 0.03
ko_coupon2Input	(str or float) the 2nd knock-out coupon	"" or 0.04
coupon2_periodInput	(str or float) the valid period of 2nd knock-out coupon	"" or 3
ko_coupon3Input	(str or float) the 3rd knock-out coupon	"" or 0.02
coupon3_periodInput	(str or float) the valid period of 3rd knock-out coupon	"" or 3
guaranteedInput	(str or float) autocall guaranteed coupon	"" or 0.01
limitedlossInput	(str or float) minimum price of put snowball	"" or 0.8
parachuteInput	(str or float) last knock-put price of parachute	"" or 0.85

A template backtesting pramater JSON file is as followed:

{ "structureInput": "classic", "codeInput": "000905.XSHG", "durationInput": 6, "start_obInput": 3, "div_couponInput": 0.06, "ko_couponInput": 0.06, "kiInput": 0.8, "koInput": 1.03, "sdateInput": "01/01/2022", "edateInput": "12/31/2022", "sdInput": "", "ko_coupon2Input": 0.04, "coupon2_periodInput": 3, "ko_coupon3Input": "", "coupon3_periodInput": "", "guaranteedInput": "", "limitedlossInput": "", "parachuteInput": "" }

1.2 Result Object

The backtesting result should be created by the corresponding Python file with following keys:

Table 2: Backtesting Result JSON

JSON Key	Context	Example
start_date	(timestamp) start date of contract	1641254400000
close	(float) close price of the start date	7354.4562
payoff	(float) payoff of the option starting on start_date	-0.1130369231
state	(str) end state of the option	"ki_noko"
actual_end_date	(timestamp) actual end date of the contract starting on start_date	1656892800000

2 Monte Carlo Page

Similarly to the backtesting file, the Monte Carlo page has two objects: "parameter" and "path". A template Monte Carlo file is as followed:

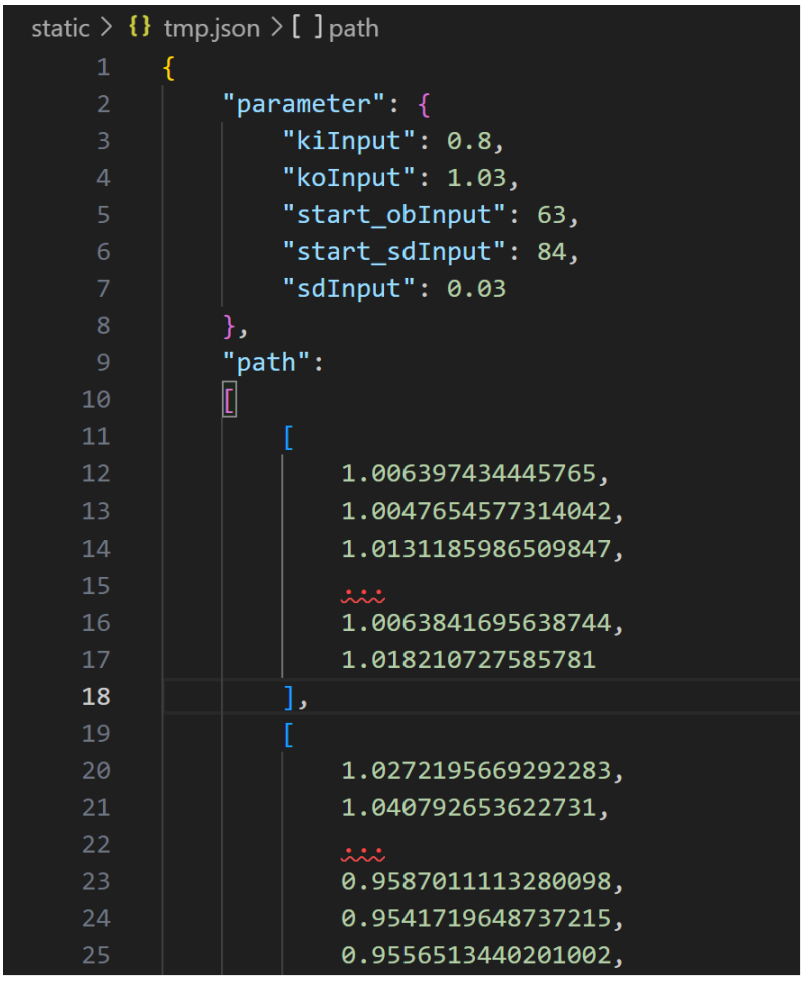

Figure 2: Monte Carlo Parameter JSON

2.1 Parameter Object

Similarly, all of the parameters of Monte Carlo are permitted to be null if needed, and the renderer process of JavaScript will automatically judge whether the inputs are valid according to the structure.

Table 3: Monte Carlo Parameter JSON

JSON Key	Context	Example
kiInput	(float) knock-in price	0.8
koInput	(float) knock-out price	1.03
start_obInput	(str or int) the 1st trading day starting observation: should be multiples of 21	"" or 63
start_sdInput	(str or int) the 1st trading day starting stepdown: should be multiples of 21	"" or 84
sdInput	(str or float) stepdown	"" or 0.03

2.2 Path Object

The path is a $100(200) \times 252$ array, where 100(200) is the number of path, and 252 is the number of trading day, with knock-out observation days of 21, 42, ...252.