

Coding task

We expect you to create a small .NET Core backend service which calculates taxes applied in different municipalities. The taxes are scheduled in time. Service should provide the API to get taxes applied in certain municipality at the given day.

Each municipality can have its own tax calculation rules. There should be 2 starting rules on how intersecting taxes could be applied. First one, is to add them up, second one is to choose the smallest period tax that covers the chosen day.

Examples:

Municipality Vilnius is using tax rule 2 and has its taxes scheduled like this:

- yearly tax = 0.2 [for period 2020.01.01-2020.12.31],
- monthly tax = 0.4 [for period 2020.05.01-2020.05.31],
- it has no weekly taxes scheduled,
- it has two daily taxes scheduled = 0.1 [at days 2020.01.01 and 2020.12.25]

Municipality Kaunas is using tax rule 1 and has its taxes scheduled like this:

- yearly tax = 0.3 [for period 2020.01.01-2020.12.31]
- monthly tax = 0.2 [for period 2020.01.01-2020.01.31]
- weekly tax = 0.1 [for period 2020.01.06-2020.01.12]
- it has no daily taxes

The result according to provided example would be:

Municipality	Date	Tax rule	Result
Vilnius	2020.01.01	2	0.1
Vilnius	2020.05.02	2	0.4
Vilnius	2020.07.10	2	0.2
Vilnius	2020.03.16	2	0.2
Kaunas	2020.01.01	1	0.5
Kaunas	2020.01.08	1	0.6
Kaunas	2020.02.10	1	0.3
Kaunas	2020.03.16	1	0.3

Technical requirements:

- Code should be flexible to change/add tax calculation rules
- Backend service should be able to use multiple data storage options. At least one data storage should be implemented.
- Backend service should have logging implemented, with an output to console
- Extra points for unit tests
- Source code should be pushed to a publicly available git repository