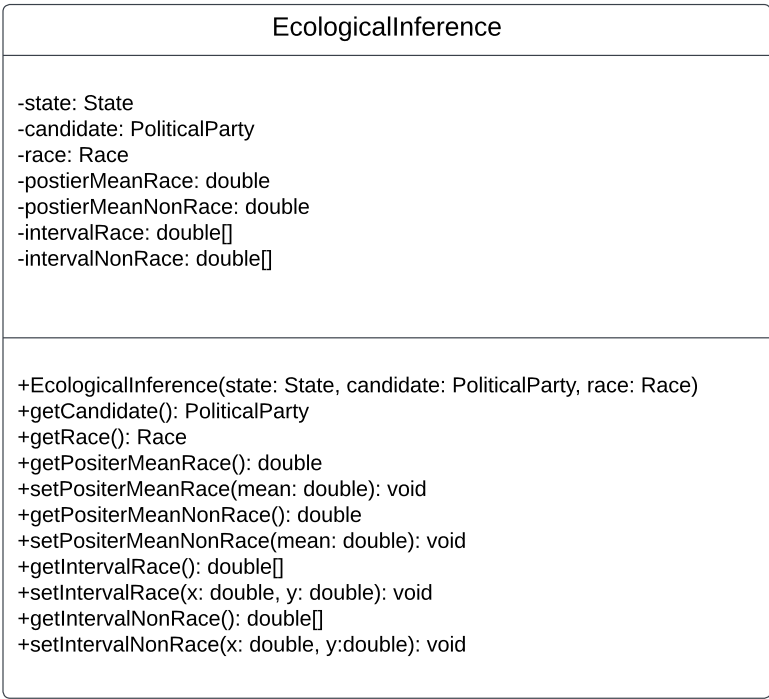


**Ecological Inference DataFrame Columns:**

- # CANDIDATE: The candidate of who's support we are analyzing
- # RACE: The race we are analyzing
- # POSTIER\_MEAN\_RACE: The posttier mean for the district-level voiting preference of race for candidate
- # POSTIER\_MEAN\_NON\_RACE: The posttier mean for the district-level voting preference of non\_race for candidate
- # INTERVAL\_RACE: The 95% equal-tailed Bayesian credible interval for district-level voting preference of race for candidate
- # INTERVAL\_NON\_RACE: The 95% equal-tailed Bayesian credible interval for district-level voting preference of non\_race for candidate



# Interface Document

## /state/coords

- brief: Gets the center coordinates of a state
- param: state
  - The state of who's coordinates to get
- returns: JSON

```
{
  "lat": 0,
  "long": 0
}
```

## /state/plan

- brief: Gets the state's enacted district plan
- param: state
  - The state of who's plan to get
- returns: JSON

```
{
  "geometries": [
    [
      [
        [0.0, 0.0],
        ...
      ],
      ...
    ],
    ... //Multiple district objects in this array for the state
  ]
}
```

## /state/heat

- brief: Gets heat map data for a race
- param: state
  - The state of who's representatives to get
- param: race
  - The race of who's heat map is being generated
- returns: JSON

```
{
  [
    {
      "district": 0,
      "color": ""
    },
    {
      "district": 1,
      "color": ""
    },
    ... //Multiple district objects to follow
  ]
}
```

## /state/reps

- brief: Gets all of the state's representatives
- param: state
  - The state of who's representatives to get
- returns: JSON

```
{
  "representatives": [
    {
      "name": "",
      "race": "",
      "district": 0,
      "party": "",
      "district": 0,
      "imageUrl": "",
      "voteMargin2020": 0
    },
    ... //Multiple representatives objects in this array for all in the state
  ]
}
```

## /state/repphoto

- brief: Gets a representative's photo
- param: state
  - The state of who's representatives to get
- param: name
  - The name of the representative of who's photo to get
- returns: The URL string of the representative's photo

## /state/dplan

- brief: Gets the state's district's enacted plan
- param: state
  - The state of who's district plan to get
- param: district
  - The district number of who's plan to get
- returns: JSON

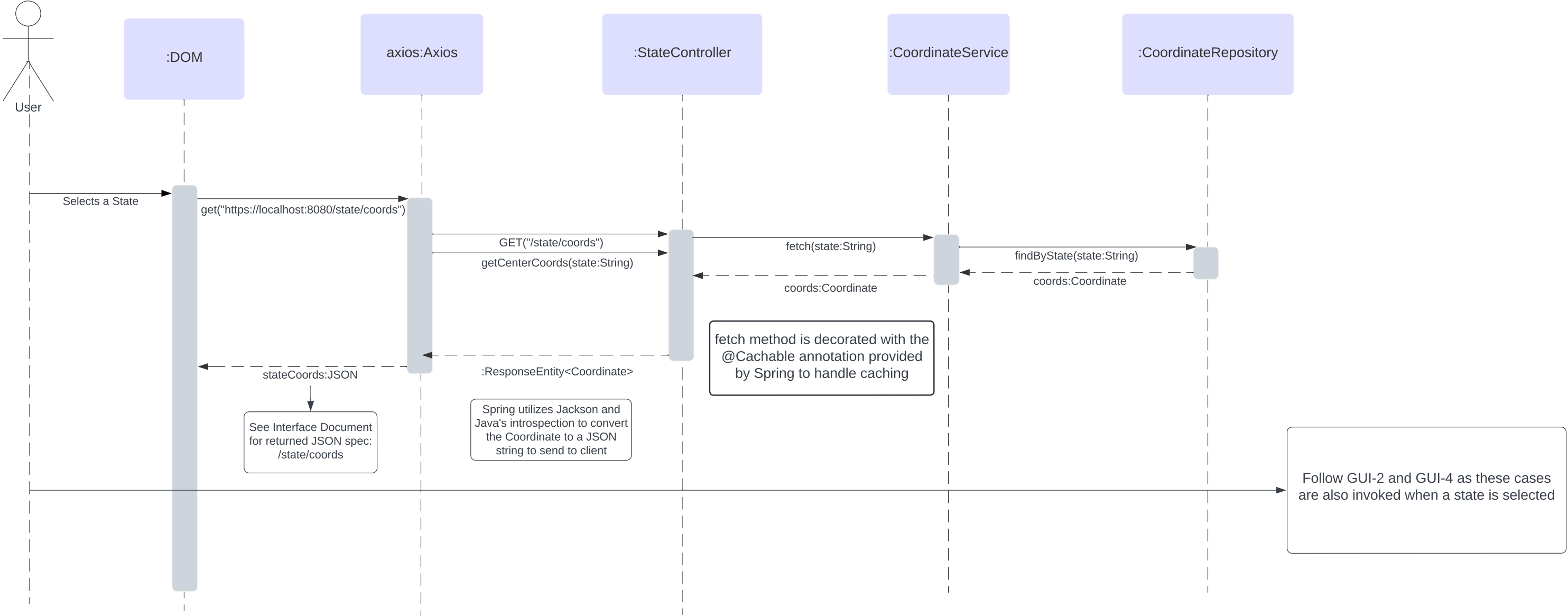
```
{
  "geometry": [
    [
      [0.0, 0.0],
      ...
    ],
    ...
  ]
}
```

## /state/all

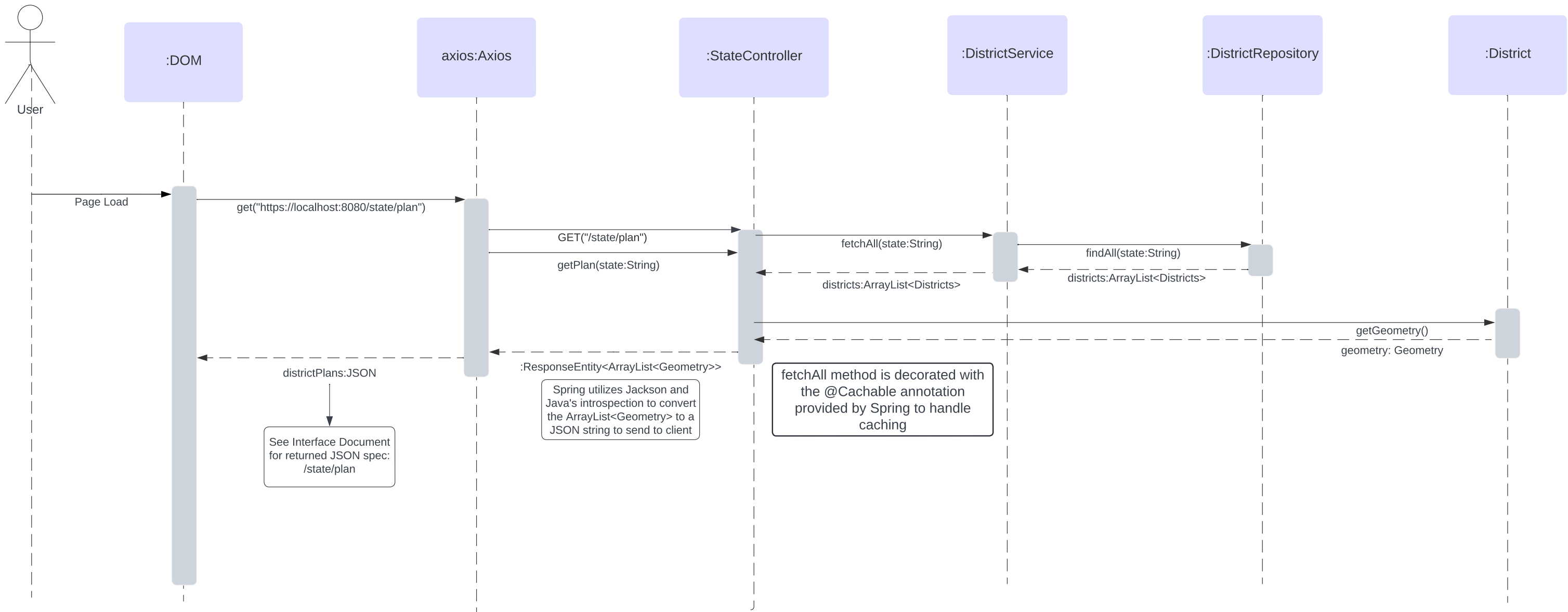
- brief: Gets all of the state's information
- param: state
  - The state of who's information to get
- returns: JSON

```
{
  "districts": [
    {
      "id": 0,
      "representative": {
        "name": "",
        "race": "",
        "party": "",
        "imageUrl": ""
      },
      "population": 0,
      "populationWhite": 0,
      "populationBlack": 0,
      "populationHispanic": 0,
      "populationAsian": 0,
      "republicanVotes2020": 0,
      "democraticVotes2020": 0,
      "winner2020": "",
      //Coordinate property excluded as not used in this use case
    },
    ... //Multiple district objects
  ]
}
```

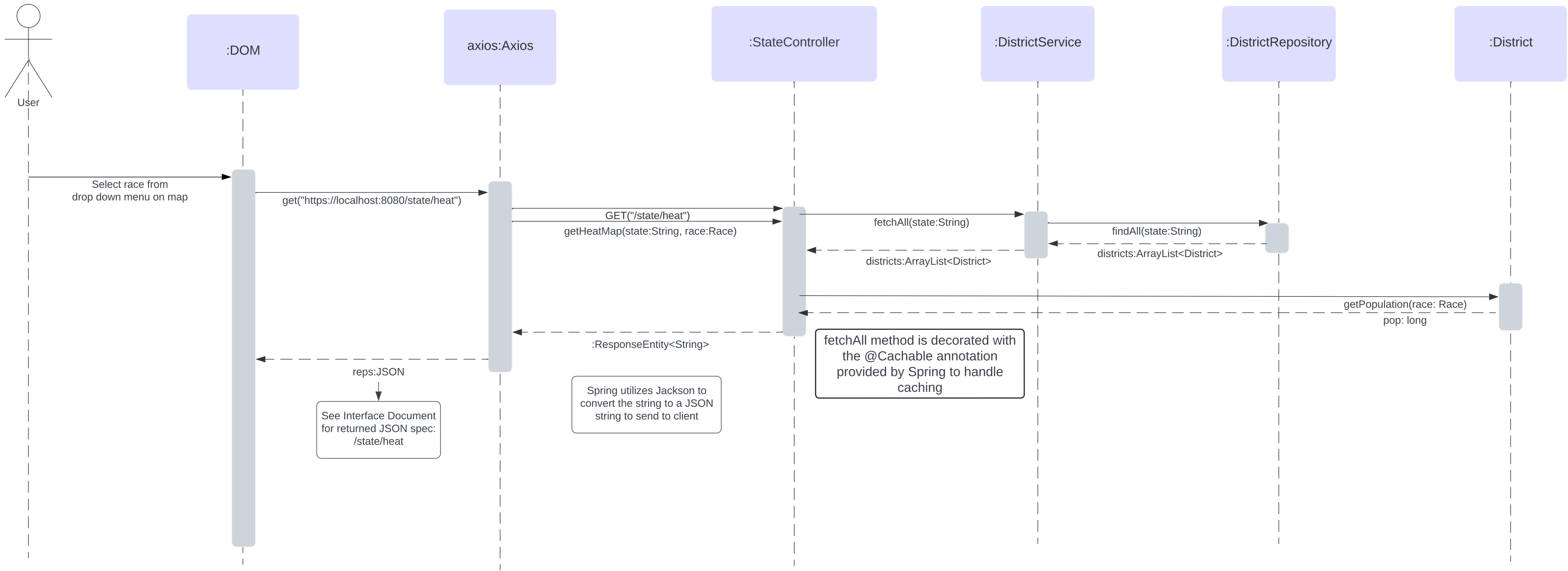
# GUI-1: Select state to display



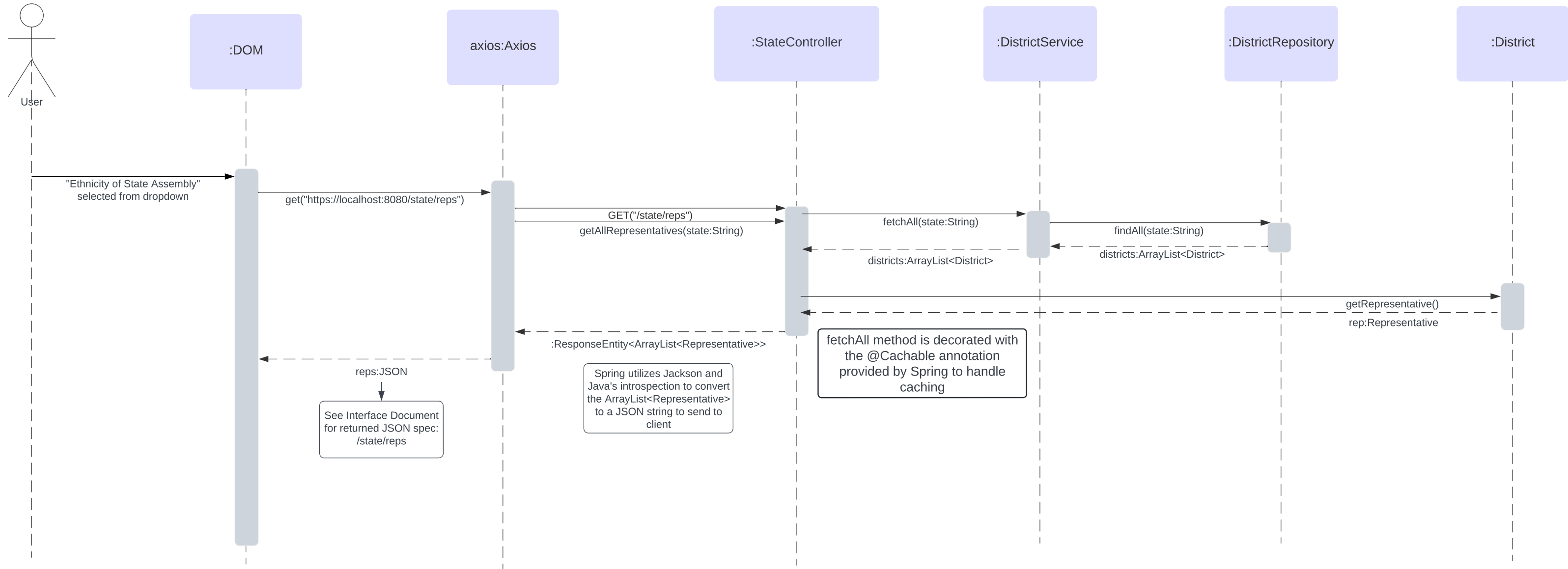
# GUI-2: Display the current district plan by default



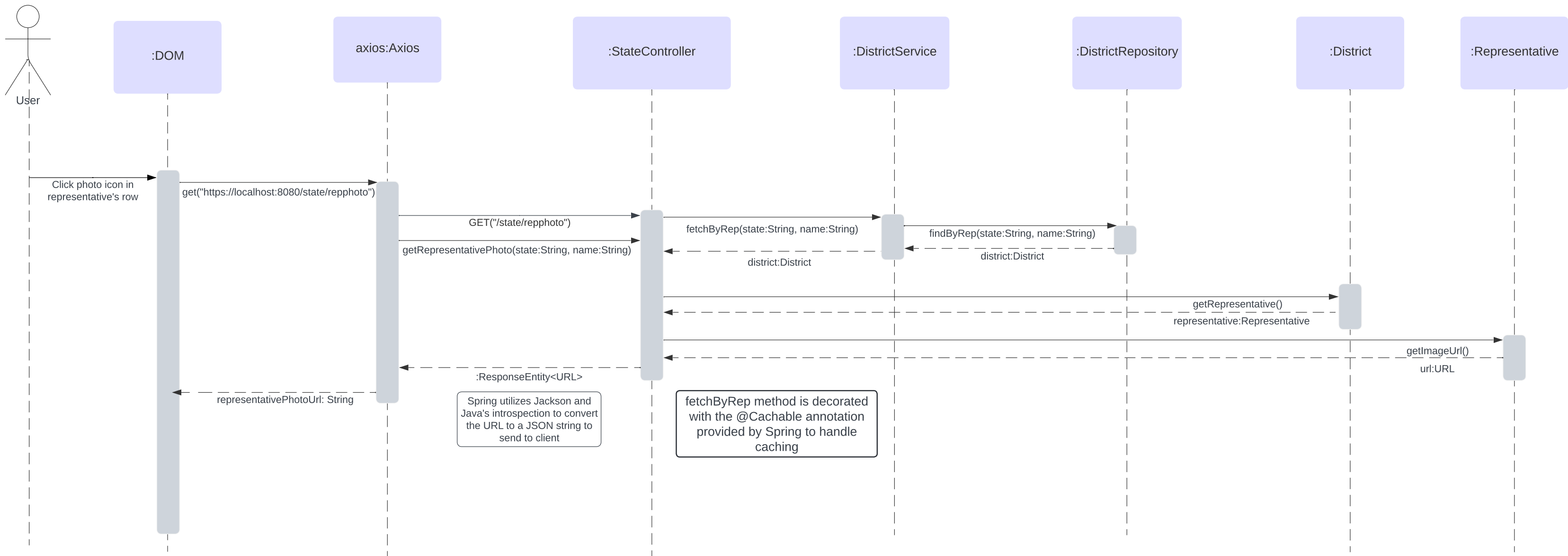
# GUI-3: Display demographic heat map



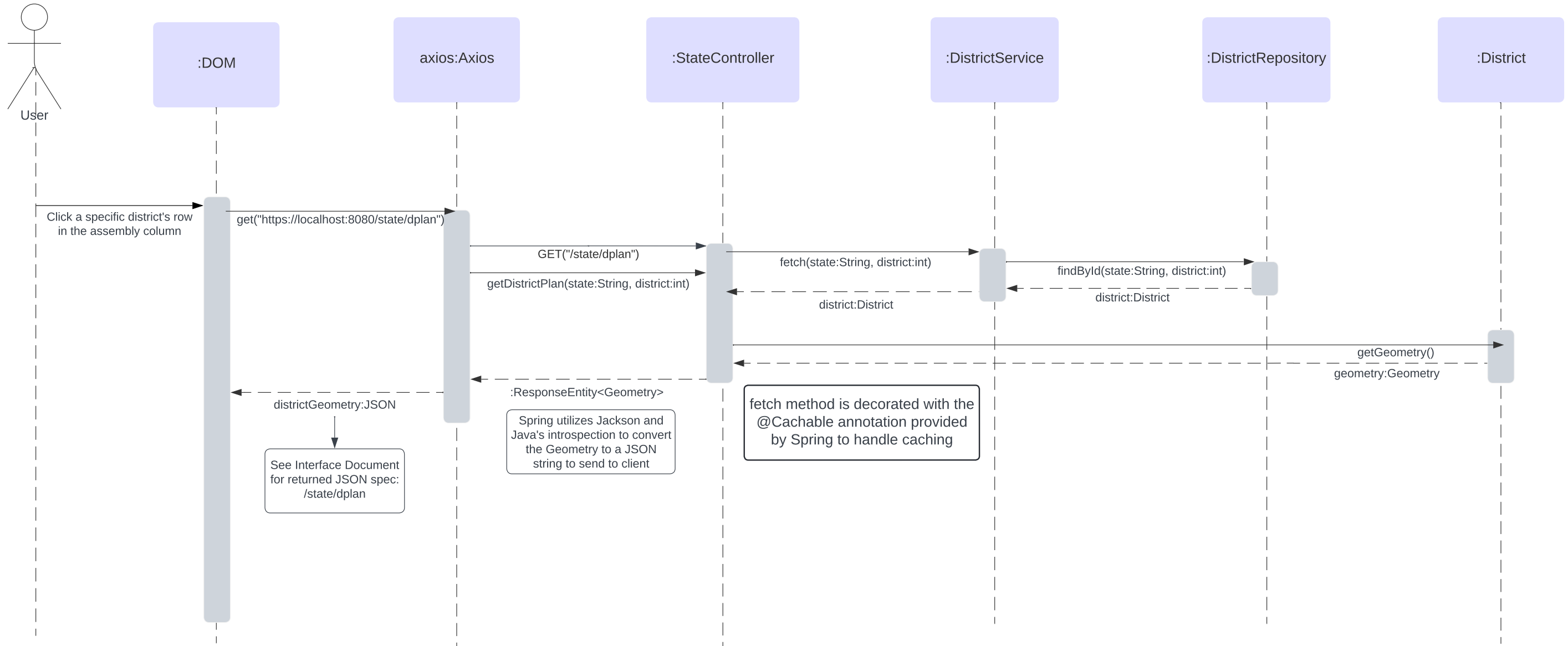
## GUI-4: Display state assembly table



**GUI-5: Display photo of district representative**



## GUI-6: Display district plan





# GUI-7: State Data Summary

