

House table

In this app you can create, get details and update house

1. Create house:
in the path: <http://localhost:8000/house>

Create A House

The form consists of three text input fields stacked vertically. The first field is labeled 'address *', the second 'currentValue *', and the third 'loanAmount *'. Each field has a small asterisk indicating it is required. Below the fields is a prominent blue button with the text 'CREATE' in white capital letters.

upon submission the form the endpoint <http://localhost:8000/api/houses> is call and create the house in the table in this function in server side ,classes/house.js:

```
//create a new house
//function get address,currentValue,loanAmount
//and return new house id or error
module.exports.createHouse=async function (address,currentValue,loanAmount){
  try{
    var risk=0
    var loanAmount=parseFloat(loanAmount)
    var currentValue=parseFloat(currentValue)
    if(loanAmount<currentValue)
      return ("loanAmount must be >=currentValue")
    risk=currentValue/loanAmount
    if (loanAmount>=currentValue+(currentValue/100)*50)
      risk+=(risk/100)*10
    const house = await DB.create({ address: address ,currentValue:currentValue,loanAmount:loanAmount,risk:risk.toFixed(3) });
    return(house.id)}
  catch(error){
    return({"error":error})
  }
}
```

2. Get house details after creation:
in the path : <http://localhost:8000/details>

House Details

Id:

177

Address:

addressd

currentValue :

5

loanAmount:

5

Risk:

1

UPDATE

the endpoint <http://localhost:8000/api/houses/specificid> is call and return the details of specific house in in this function in server side , classes/house.js:

```
//get house details
//function get id and return house object or error
module.exports.getHouse=async function (id){
  try{
    var house= await DB.findAll({
      where: {
        id: id
      }
    });
    return(house)}
  catch(error){
    return({"error":error})
  }
}
```

3. Update house in the path: <http://localhost:8000/update>

Update A House

id

177

address

addressd

currentValue

5

loanAmount*

5

risk

1

UPDATE

upon submission the form the endpoint

<http://localhost:8000/api/houses/specificid> is call and update the specific house in the table in this function in server side ,classes/house.js:

```
//update house
//function get :id,address,currentValue,loanAmount
//update address,currentValue,loanAmount attributes and calculate rhe risk attribute
//and return an array of affected rows or error
module.exports.updateHouse=async function (id,address,currentValue,loanAmount){
  try{
    var risk=0
    var loanAmount=parseFloat(loanAmount)
    var currentValue=parseFloat(currentValue)
    if(loanAmount<currentValue)
      return ("loanAmount must be >=currentValue")
    risk=currentValue/loanAmount
    if (loanAmount>=currentValue+(currentValue/100)*50)
      risk+=(risk/100)*10
    var update_row = await DB.update({ address: address ,currentValue:currentValue,loanAmount:loanAmount,risk:risk.toFixed(3) }, {
      where: {
        id: id
      }
    });
    return(update_row)}
  catch(error){
    return({"error":error})
  }
}
```