API Insurance Claim and Policy Report

API Information

Title	Claim
Descriptio n	API responsible for managing the smart contract for claim creation and registration (Claim Factory) and the reports (Policy Report Factory) on the blockchain. It also conducts claims consultations and reports and changes in the status of both.

Input Parameters

Below is the list of parameters expected for the API to perform its main function. This input information will be provided by the platform that will consume this API - so any error in sending the parameters will directly affect the functioning of the same.

Content type: `application/json`

insurance-claim

Method	Summary	Description	Mandatory ?	Input parameter
POST/insurance-claim	Generate claim	Function responsible for generating claims	Yes	insurance-claim data { "privateKey": "string", "mutual": ["string"], "apoliceData": { "apolice": 0, "startValidity": "2020-04-06", "endValidity": "2020-04-06", "apoliceStatus": 0 }, "claimInformation": { "claim": 0, "claimStatus": "string" }, "conductorData": { "nameComplete": "string", "dateOfBirth": "2020-04-06", "maritalStatus": 0, "cnpjCpf": 0, "gender": 0, "profession": "string", "cnh": 0, "categoryCnh": "string", "dateValidateCnh": "2020-04-06" }, "occurenceData": { "claimNumber": 0, "dateOccurrence": "2020-04-06", "timeOccurence": "string", "placeOccurence": "string", "policeReport": 0, "protocolPoliceReport": 0, "conductorGuiltyOccurrence": 0, "ocurrenceDescription": 0, "victims": 0, "damageVictims": "string", "damageVehicle": "string" }, "thirdPartyData": { "involvement3rd": 0, "howManyInvolvement": 0,

				"licensesPlates": ["string"], "damageCaused": "string" }, "vehicleData": { "vehicleType": "string", "maker": "string", "model": "string", "numerSlides": 0, "yearManufacture": 0, "yearModel": 0, "licensePlate": "string", "chassis": "string", "renavam": "string", "fuel": 0 } }

PUT/insurance-claim	Update Sinister status	Function to update the status of the claim	Yes	`privateKey`
				`address`
				`status`

POST/insurance-claim/consult-claims	Consult Claim	Consult Claim	Yes	`privateKey` `contractAddress`
POST/insurance-claim/consult-mutual-claims	Insurance Claims	check for Insurance company Claims	Yes	`privateKey` `multualAddress`

policy-report

Method	Summary	Description	Mandatory?	Input parameter

POST/policy-report	Generate Report	Generate Claim Report	Yes	˅ policy-report data
				{
				"privateKey": "string",
				"mutual": [
				"string"
				],
				"occurrenceData": {
				"claimNumber": "string",
				"dateOcurrence": "2020-04-06",
				"timeOcurrence": "2020-04-06T22:02:14.153Z",

				"placeOcurrence": "string",
				"policeReport": "string",
				"protocolPoliceReport": "string",
				"conductorGuiltyOccurrence": "string",
				"ocurrenceDescription": "string",
				"victims": "string",
				"damageVictims": "string",
				"damageVehicle": "string"
				},
				"assistanceData": {
				"cnpjCpf": "string",
				"nameComplete": "string"
				},
				"technicalReport": {
				"reportNumber": "string",
				"reportStatus": "string",
				"damageComponents": [
				"string"
				]
				},
				"manPower": "string",
				"technicalAdvice": "string"
				}

PUT/policy-report	Update Status Report	Update Status Report	Yes	`privateKey`
				`address`
				`status`

POST/policy-report/consult-mutual-policies-report	Consult insurance reports	Lists the reports belonging to an insurance company	Yes	`privateKey` `contractAddress`

POST/policy-report/consult-policy-report	Check report	Check report	Yes	privateKey multualAddress

administrativo insurance-claim e policy-report

Method	Summary	Description	Mandatory?	Input parameter
POST/role/admin	Add admin	Add an admin to the smart contract onlyAdmin	Yes	contract address privateKey
DELETE/role/admin	Remove admin	Remove an admin from the smart contract onlyAdmin	Yes	contract address privateKey
GET/role/admin/{contract}/{address}	Check admin	Check if an address is an admin in that smart contract	Yes	contract address
POST/role/mutual	Add insurance company	Adds an address to the role of insurance in the smart contract onlyAdmin	Yes	contract address privateKey
DELETE/role/mutual	Remove insurance company	removes an address from the role of insurance in the smart contract onlyAdmin	Yes	contract address privateKey
GET/role/mutual/{contract}/{address}	Check insurance company	Returns whether the address has the role of insurance company in the smart contract	Yes	contract address
POST/lifecycle/pause	Pause the contract	If the smart contract is in unpause status, this method pauses the smart contract by changing the status to pause.	Yes	contract privateKey

POST/lifecycleq/unpause	Activate the contract	If the smart contract is in the pause status this method reactivates the smart contract by changing the status to unpause	Yes	`contract` `privateKey`
GET//lifecycle/paused/{contract}	Contract status	Checks whether the smart contract is in pause or unpause status	Yes	`contract`

List of smart contract functions called by API methods *Claim*

Method	**Function**	**Description**
POST/insurance-claim	`generateClaim`	Function responsible for generating Claim
PUT/insurance-claim	`updateStatusClaim`	Function to update the status of the claim

POST/insurance-claim/consult-claims	`consultApolice`
	`consultVehicle`
	`consultVehicleLicence`
	`consultConductor`
	`consultOcurrence`
	`consultOcurrenceReport`
	`consultThirdParty`
	`consultClaimInformation`
	`consultClaimMutual`
		Consultation of claims

POST/insurance-claim/consult-mutual-claims	`consultMutuaClaims`	Consultation of claims of an insurance company
POST/policy-report	`generateReport`	Generate Claim Report
PUT/policy-report	`updateStatusReport`	Update of the Status of the Report
POST/policy-report/consult-mutual-policies-report	`consultMutualPoliciesReport`	Lists the reports belonging to an insurance company

POST/policy-report/consult-policy-report	`consultReportMutual`
	`consultReportOcurrenceData`
	`consultReportAssistanceData`
	`consultReportTechnicalReport`
	`consultReport`
		Consult report information

POST/role/admin	`addAdmin`	Adds an administrator to the smart contract `onlyAdmin`
DELETE/role/admin	`removeAdmin`	Removes an administrator from the smart contract `onlyAdmin`
GET/role/admin/{contract}/{address}	`isAdmin`	Consult if an address is admin in the smart contract
POST/role/mutual	`addInsuranceCompany`	Adds a mutual to the smart contract`onlyAdmin`
DELETE/role/mutual	`removeInsuranceCompany`	Removes an insurance company from the smart contract`onlyAdmin`
GET/role/mutual/{contract}/{address}	`isInsuranceCompany`	Consult if an address is an insurance company in the smart contract
POST/lifecycle/pause	`pause`	Pauses smart contract `onlyAdmin`
POST/lifecycleq/unpause	`unpause`	Activates smart contract `onlyAdmin`
GET/lifecycle/paused/{contract}	`isPaused`	Returns the current pause / unpause status

Output Parameters

Below is the list of parameters returned:

Method	**Output Parameter**
POST/insurance-claim	transaction link Blockexplorer
PUT/insurance-claim	transaction link Blockexplorer
POST/insurance-claim/consult-claims	Claim information value

POST/insurance-claim/consult-mutual-claims	Lists address contracts claims and owners { "success": true, "code": "100-1000", "message": "sucess", "data": [["0x139d6E02A3fF517b11971c6c49f75BfD33741ee6", "0x8f03bE3AD8971202da1aEAB4dB0ee78682cabB6b", "0xe421e1CC7A39F84598d1B85d0858c6113BB28D37", "0xAbacaB210CE4A502Efc5a88885c9a95094821878", "0x995967Cc0783E75a6EdCE62d763D15088C06e922"], ["0x17cA6A08758F4A078B9c53ca25E6F6736dF34094", "0x17cA6A08758F4A078B9c53ca25E6F6736dF34094", "0x275d671F09697b46EE82238ede224fC89eA26365", "0x275d671F09697b46EE82238ede224fC89eA26365", "0x275d671F09697b46EE82238ede224fC89eA26365"]] }
POST/policy-report	transaction link Blockexplorer
PUT/policy-report	transaction link Blockexplorer
POST/policy-report/consult-mutual-policies-report	Lists address contract police report and owner { "success": true, "code": "100-1000", "message": "sucess", "data": [["0x139d6E02A3fF517b11971c6c49f75BfD33741ee6", "0x8f03bE3AD8971202da1aEAB4dB0ee78682cabB6b", "0xe421e1CC7A39F84598d1B85d0858c6113BB28D37", "0xAbacaB210CE4A502Efc5a88885c9a95094821878", "0x995967Cc0783E75a6EdCE62d763D15088C06e922"], ["0x17cA6A08758F4A078B9c53ca25E6F6736dF34094", "0x17cA6A08758F4A078B9c53ca25E6F6736dF34094", "0x275d671F09697b46EE82238ede224fC89eA26365", "0x275d671F09697b46EE82238ede224fC89eA26365", "0x275d671F09697b46EE82238ede224fC89eA26365"]] }
POST/policy-report/consult-policy-report	Report information value
POST/role/admin	transaction link Blockexplorer

DELETE/role/admin	transaction link Blockexplorer
GET/role/admin/{contract}/{address}	{ "success": true, "code": "100-2030", "description": "success", "data": { "isAdmin": true, "contract": "insurance-claim" } }
POST/role/mutual	transaction link Blockexplorer
DELETE/role/mutual	transaction link Blockexplorer
GET/role/mutual/{contract}/{address}	{ "success": true, "code": "100-2032", "description": "success", "data": { "isMutual": true, "contract": "insurance-claim" } }
POST/lifecycle/pause	transaction link Blockexplorer
POST/lifecycleq/unpause	transaction link Blockexplorer
GET//lifecycle/paused/{contract}	contract / status: <pause/unpause>

Messages

code	100-1000	Success! <response>
code	100-1010	Return list of errors
code	100-1013	Insurance Company has no claim.
code	100-1014	Invalid private key <privetKey>
code	100-1017	Status: <pause/unpause> <nameContract>
code	100-1020	Smart Contract address is invalid.

code	100-1030	Smart Contract address is empty.
code	100-1031	Transaction subscriber has to be Insurance Company.
code	100-1033	Smart contract <nameSmartContract> is <pause/unpause>
code	100-1040	Invalid address Insurance Company
code	100-1050	Empty address Insurance Company
code	100-2010	Transaction to add admin <nameContract>!
code	100-2011	Transaction to add mutual <nameContract>!
code	100-2020	Transaction to remove admin <nameContract>!
code	100-2021	Transaction to remove mutual <nameContract>!
code	100-2030	Is Admin = true
code	100-2031	Is admin = false
code	100-2032	Is Insurance Company= true

code	100-2033	Is Insurance Company = false
code	100-2100	Registered Report <address>
code	100-3011	Transaction update status insurance claim
code	100-3021	Transaction update status policy report.
code	100-3100	Transaction Failed, address does not have permission.
code	100-7002	Wallet address is invalid.
code	100-7017	Transaction register insurance claim!

Data dictionary

insurance-claim

Category	**Variable**	**Description**	**Type**	**Example**
privateKey		Private key of transaction signer (msg.sender)	`string`	A9AC29CBEB110215AE6D5AF5D8731A848160A3A7DABEF198B832A38D4B4385E2
mutual				
	`wallets`	Numbers of the wallets of each insurance	`string`	[0x09c15427Fed859ed46AFFB996bCd62f3b9180137]

		company belonging to the claim.		
apoliceData				
	apolice	Policy identification (internal)	uint256	123456
	start_validity	Effective date	uint256	01/01/2020
	end_validity	Effective end date	uint256	02/01/2020
	apolice_status	Policy status	uint256	1
claimInformation				
	claim	Claim identification (internal)	string	123456
	claimStatus	Claim status	uint256	1
conductorData				
	name_complete	Driver name	string	client
	date_of_birth	Date of birth	string	01/01/2020
	marital_status	Marital status	uint256	1
	cnpj_cpf	CPF / CNPJ	string	11111111111

	gender	Gender	`uint256`	1
	profession	Profession	`uint256`	dev
	cnh	Driver license	`uint256`	1234567890
	category_cnh	License category	`string`	b
	date_validate_cnh	License expiration date	`string`	01/03/2020
ocurrenceData				
	claim_number	Claim number (internal)	`uint256`	123456
	date_occurrence	Date of occurrence of the claim	`string`	01/10/2020
	time_occurrence	Time of occurrence of the claim	`string`	13
	place_occurrence	Place	`string`	place
	police_report	Police report	`uint256`	678767
	protocol_police_report	Police report protocol	`uint256`	1
	conductor_guilty_occurrence	Driver at the time of occurrence	`uint256`	1

	ocurrence_description	Occurrence description	`string`	adc
	victims	Number of victims	`uint256`	1
	damage_victims	Description of the damage caused to the victims	`uint256`	1
	damage_vehicle	Description of the damage caused to the vehicle	`uint256`	1
thirdPartyData				
	involvement_3rd		`uint256`	2
	how_many_involvement		`uint256`	3
	licenses_plates		`string[]`	[aaa1234]
	damage_caused		`string`	none
vehicleData				
	vehicle_type	Vehicle type	`uint256`	1
	maker	Maker	`uint256`	4
	model	Model	`uint256`	19
	numer_slides		`uint256`	5

	year_manufacture	Year of manufacture	`uint256`	2010
	year_model	Model year	`uint256`	2011
	license_plate	License plate	`string`	1
	chassis	Chassis	`string`	11111A1111
	renavam	Renavan	`string`	11111N1111
	fuel	Fuel type	`uint256`	1

policy-report

Category	**Variable**	**Description**	**Type**	**Example**
privateKey		Private key of transaction signer (msg.sender)	`string`	A9AC29CBEB110215AE6D5AF5D8731A848160A3A7DABEF198B832A38D4B4385E2
mutual				
	wallets	Numbers of the wallets of each insurance company belonging to the report.	`string`	[0x09c15427Fed859ed46AFFB996bCd62f3b9180137]
occurrenceData				
	claimNumber	Claim identification (internal)	`string`	321123
	dateOcurrence	Date of occurrence	`uint256`	1585084539000
	timeOcurrence	Time of occurrence	`uint256`	1585084539000

	placeOcurrence	Place of occurrence	string	Avenida Paulista1392
	policeReport	Police report	uint256	121
	protocolPoliceReport	Police report protocol	uint256	321344
	conductorGuiltyOccurrence	Driver at the time of occurrence	uint256	1
	ocurrenceDescription	Occurrence description	string	dsad11
	victims	Number of victims	uint256	1
	damageVictims	Description of the damage caused to the victims	string	Houve danos das vitimas descrição
	damageVehicle	Description of the damage caused to the vehicle	string	Relato dos danos causados no veiculo
assistanceData				
	cnpjCpf	CPF/CNPJ	string	99999999999
	nameComplete	Full name of technical assistance	string	Assistencia tecnica
technicalReport				
	reportNumber	Report identification (internal)	string	252415
	reportStatus	Report status	uint256	2

	damageComponents	Damaged components	`string[]`	[2, 1, 32]
	manpower	Description of labor	`string`	Descrição da mão de obra
	technicalAdvice	Technical opinion report	`string`	Relato do parecer técnico

Implementation

Requirements

- NPM >=6.13.4
- GIT >=2.21.1
- NODE.JS >= 10.0.0

Clone repository

You need to have ssh configured, with the access and key you can clone the repository from the terminal:

```
git clone git@bitbucket.org:janusplatform/Claims.git
```

Install dependencies

In the terminal access the root folder of the repository that was cloned run the command to install the dependencies:

```
npm install
```

All dependencies needed to run the API will be installed in your directory.

Create and configure the .env file

In order to run the API, it is necessary to create the .env file at the root of the directory Clone the .env.example file with the following specifications:

```
MNENOMIC = // Your metamask's recovery words
INFURA_API_KEY = // Your Infura API Key after its registration
NETWORK_ID = // 1-Mainnet 3-Ropsten 4-Rinkeby 42-Kovan 1001-Development
TOKEN = // PDBToken Address
```

After creating and configuring .env, your directory is ready to run the API.

Run the nodemon on localhost

In the terminal access the root folder of the repository that was cloned execute the command to run the API:

```
npm run start
```

After executing this command the nodemon will be initialized and the API will be working at the address http://localhost:3000.