Warning

Collapse

⚠ Swagger Warning

```
Definition is not used: #/definitions/ModelAndView
```

Jump to line **458**

Details

```
▶ Object
```

BetRoyale api documentation

Operations pertaining to bet with another player. the data is stored in a db and in a blockchain

Version 1.0.0

License
Apache 2.0 (http://www.apache.org/licenses/LICENSE-2.0.html)

Paths

`/api/bet`

GET /api/bet

BetRoyale API

Summary

This operation pertains to get all bets stored in the db

Responses

Code	Description	Schema
200	Successfully retrieved the bets	⇄ ▾ **Iterable«Bet» {** }
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

POST /api/bet

BetRoyale API

Summary

Add a bet, this operation will persist the bet in the db and create the transactions resulting from the bet. At end a block is mined

Parameters

Name	Located in	Description	Required	Schema
bet	body	Bet between two players	Yes	⇄ ▾**Bet {** amount: number id: string player1: ▸**User { }** player2: ▸**User { }** }

Responses

Code	Description	Schema
200	Successfully added in the db and mined	⇄ ▾**ResponseEntity {** body: ▸ **{ }** statusCode: string statusCodeValue: integer }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	The bet was not persisted and not mined	

Try this operation

/api/chain

GET /api/chain

BetRoyale API

Summary

This operation pertains to get the entire blockchain

Responses

Code	Description	Schema
200	Successfully retrieved the blockchain	⇄ ▾ **BlockChainDTO {** chain: ▸ **[]** length: integer }
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/api/mine

POST /api/mine

BetRoyale API

Summary

This operation pertains mine a new block with all actual transactions. It's only for test purpose

Responses

Code	Description	Schema
200	Successfully retrieved the users	⇄ ▾ **Block {** id: integer nonce: integer previousHash: string timeStamp: integer transactions: ▸ **[]** }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/api/sampleBetData

GET /api/sampleBetData

BetRoyale API

Summary

This operation pertains to get sample of bet data. The data aren't persisted It's only for test purpose

Responses

Code	Description	Schema
200	Successfully returned the sample data	⇄ ▾ Iterable«Bet» { }
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/api/transactions

POST /api/transactions

BetRoyale API

Summary

This operation pertains to add a transaction for the next block mined. It's only for test purpose

Parameters

Name	Located in	Description	Required	Schema
transaction	body	Transaction to add in the next block	Yes	⇄ ▾ **Transaction {** amount: number recipient: string sender: string }

Responses

Code	Description	Schema
200	Successfully retrieved the users	⇄ string
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/api/user

GET /api/user

BetRoyale API

Summary

This operation pertains to get all users stored in the db

Responses

Code	Description	Schema
200	Successfully retrieved the users	⇄ ▾ **Iterable«User» {** }
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

POST /api/user

BetRoyale API

Summary

Add a user, this operation will persist the user in the db

Parameters

Name	Located in	Description	Required	Schema
user	body	User to add in the DB	Yes	▾ User { id: string, pseudo: string, tag: string }

Responses

Code	Description	Schema
200	Successfully added in the db	▾ User { id: string, pseudo: string, tag: string }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/api/v1/chain

GET /api/v1/chain

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ▶ **[]** length: integer }
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

HEAD /api/v1/chain

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ▶ **[]** length: integer }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

POST /api/v1/chain

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ▶ **[]** length: integer }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

PUT /api/v1/chain

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ▶ **[]** length: integer }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

DELETE /api/v1/chain

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ▶ **[]** length: integer }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

`OPTIONS /api/v1/chain`

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ►**[]** length: integer }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

PATCH /api/v1/chain

BetRoyale API with another version of proof of work

Summary

getChain

Responses

Code	Description	Schema
200	OK	⇄ ▼ **BlockChainDTOV1 {** chain: ▶ **[]** length: integer }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

/api/v1/mine

POST /api/v1/mine

BetRoyale API with another version of proof of work

Summary

createBlock

Responses

Code	Description	Schema
200	OK	⇄ ▾ **BlockV1 {** id: integer nonce: integer previousHash: string timeStamp: integer transactions: ▸ **[]** }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/api/v1/transactions

POST /api/v1/transactions

BetRoyale API with another version of proof of work

Summary

createTransaction

Parameters

Name	Located in	Description	Required	Schema
transaction	body	transaction	Yes	⇄ ▾ **Transaction** { amount: number, recipient: string, sender: string }

Responses

Code	Description	Schema
200	OK	⇄ string
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

/error

GET /error

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

HEAD /error

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

POST /error

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

PUT /error

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Try this operation

`DELETE /error`

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

`OPTIONS /error`

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

PATCH /error

basic-error-controller

Summary

error

Responses

Code	Description	Schema
200	OK	⇄ ▼ { }
204	No Content	
401	Unauthorized	
403	Forbidden	

Try this operation

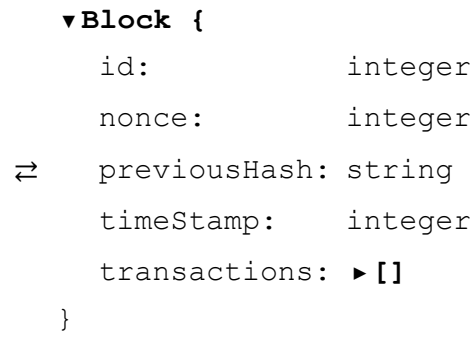
Models

Bet

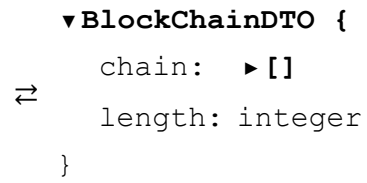
⇄

```
▼Bet {
  amount:  number
  id:      string
  player1: ▶User { }
  player2: ▶User { }
}
```

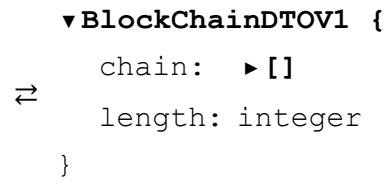
Block



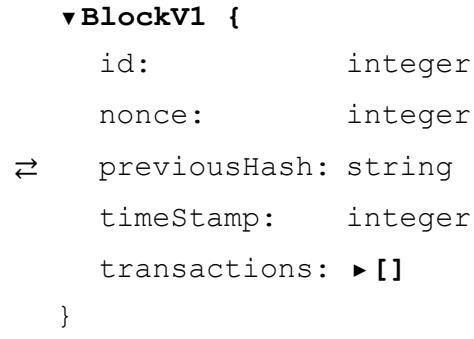
BlockChainDTO



BlockChainDTOV1



BlockV1



Iterable«Bet»

⇄ }

Iterable«User»

⇄ ▾ **Iterable«User» {**
}

ModelAndView

▾ **ModelAndView {**
  empty: boolean
  model: ▸ **{ }**
  modelMap: ▸ **{ }**
⇄ reference: boolean
  status: string
  view: ▸ **View { }**
  viewName: string
}

ResponseEntity

▾ **ResponseEntity {**
  body: ▸ **{ }**
⇄ statusCode: string
  statusCodeValue: integer
}

Transaction

▾ **Transaction {**
  amount: number
⇄ recipient: string
  sender: string
}

User

```
▾User {
  id:     string
⇄ pseudo: string
  tag:    string
}
```

View

```
▾View {
⇄ contentType: string
}
```

Try a sample here

API Design First Workflow

Easily create API specifications

Write API specs in YAML...

...Preview documentation in Swagger