ID	Summary	Test data (creds, pre condition)	Description	Steps	Expected result	Status	Comment
/bitcoin/transactions/send							
1	Check request with valid data	Valid wallet with non empty balance on it	Trying to send request with full walid data and HTTP method	1) Fill "/bitcoin/transactions/send" in endpoint field	1) Endpoint was filled	Failed	
				2) Select PUT method	2) Method was chosen		

				3) Fill in the request body with valid data in this format :
				{
				"config": {
				"address": "xxx",
				"privateKey": "xxx",
				"wif": true
				},
				"data": [
				{
				"address": "xxx",
				"amount": "xxx"
				}
				]
				}
					3) Data were filled		

				4) Send the request	4) Response code 200 OK, the thatsaction hesh was returned in format:
					{
					"txHash":
					"6e977607665c3c9d341bbbf4008a6e1136ac095eaf940d105a908130f2702080"
					}		

				5) Check that transaction hash is valid, f.e. here	5) Transaction hash is valid, transaction data is correct		
2	Check request with invalid sender address		Trying to send transaction from invalid sender address	1) Fill "/bitcoin/transactions/send" in endpoint field	1) Endpoint was filled	Passed	
				2) Select PUT method	2) Method was chosen		

				3) Fill the request body with random config.address in this format :
				{
				"config": {
				"address": "xxx",
				"privateKey": "xxx",
				"wif": true
				},
				"data": [
				{
				"address": "xxx",
				"amount": "xxx"
				}
				]
				}
					3) Data were filled		

				4) Send the request	4) 400 error was returned with "msg": "Invalid address" message		
3	Check request with invalid receiver address		Trying to send transaction from valid sender address to invalid receiver address	1) Fill "/bitcoin/transactions/send" in endpoint field	1) Endpoint was filled	Failed	Such an expected result in step 4 is formed according to the behavior in test case № 2
				2) Select PUT method	2) Method was chosen		

				3) Fill the request body with valid config.addres and random data.address in this format :
				{
				"config": {
				"address": "xxx",
				"privateKey": "xxx",
				"wif": true
				},
				"data": [
				{
				"address": "xxx",
				"amount": "xxx"
				}
				]
				}
					3) Data were filled		

				4) Send the request	4) 400 error was returned with "msg": "Invalid address" message		
				1) Fill "/bitcoin/transactions/send" in endpoint field	1) Endpoint was filled		
				2) Select PUT method	2) Method was chosen		

4	Check request with invalid sender private key		Trying to send transaction from valid sender address but with invalid private key	3) Fill the request body with valid config.addres and invalid private key :
				{
				"config": {
				"address": "xxx",
				"privateKey": "xxx",
				"wif": true
				},
				"data": [
				{
				"address": "xxx",
				"amount": "xxx"
				}
				]
				}
					3) Data were filled	**Failed**	Such an expected result in step 4 is formed according to the behavior in test case № 2

				4) Send the request	4) 400 error was returned with "msg": "Invalid address" message		
5	Check request with wallet with no coins		Trying to send transaction from valid wallet with no coins in it	1) Fill "/bitcoin/transactions/send" in endpoint field	1) Endpoint was filled	**Passed**	
				2) Select PUT method	2) Method was chosen		

				3) Fill the request body with valid empty wallet in config.addres in such format:
				{
				"config": {
				"address": "xxx",
				"privateKey": "xxx",
				"wif": true
				},
				"data": [
				{
				"address": "xxx",
				"amount": "xxx"
				}
				]
				}
					3) Data were filled		

				4) Send the request	4) 500 error was returned with "msg": "No available unspent txs" message		
/litecoin/transactions/send							
6	Check request with valid data	Valid wallet with non empty balance on it	Trying to send request with full walid data and HTTP method	1) Fill "/litecoin/transactions/send" in endpoint field	1) Endpoint was filled	**Failed**	
				2) Select PUT method	2) Method was chosen		

				3) Fill the request body with random config.address data in this format :
				{
				"config": {
				"address": "xxx",
				"privateKey": "xxx",
				"wif": true
				},
				"data": [
				{
				"address": "xxx",
				"amount": "xxx"
				}
				]
				}
					3) Data were filled		

				4) Send the request	4) Response code 200 OK, the thatsaction hesh was returned in format:
					{
					"txHash":
					"6e977607665c3c9d341bbbf4008a6e1136ac095eaf940d105a908130f2702080"
					}		

				5) Check that transaction hash is valid, f.e. here	5) Transaction hash is valid, transaction data is correct		