# Project Purpose:

The purpose of the Insurance Smart Contract project is to leverage blockchain technology to build a decentralized, transparent, and automated insurance system. Traditional insurance systems suffer from inefficiencies, lack of transparency, high administrative costs, and delays in claim processing. By implementing smart contracts on a blockchain (e.g., Ethereum), this project aims to address these issues by automating policy issuance, premium payments, claim submissions, and claim payouts. The system will provide policyholders with secure and tamper-proof insurance services while reducing fraud and human error. Ultimately, the smart contract enhances trust, ensures swift claim processing, and increases customer satisfaction by delivering efficient, transparent, and reliable insurance services.

# Synopsis:

This project involves the development of a **decentralized insurance management system** using smart contracts deployed on the Ethereum blockchain. The system automates key processes in the insurance lifecycle, such as:

* **Policy issuance**: Insurers can issue policies, define premiums, coverage amounts, and policy duration.
* **Premium payments**: Policyholders can pay their premiums directly through the smart contract, ensuring the policy remains active.
* **Claim submission**: Policyholders can submit claims with the claim amount and reason, and the contract will validate whether the claim meets the policy requirements.
* **Claim approval and payout**: Insurers can approve valid claims, and the smart contract ensures swift payout directly to the policyholder’s wallet.

The **immutable** and **transparent nature** of the blockchain ensures that all transactions and updates (such as policy issuance, premium payment, and claim status) are recorded in an open, tamper-proof ledger. This guarantees the security, integrity, and transparency of the insurance process. By eliminating the need for intermediaries and automating manual processes, the smart contract increases efficiency, reduces costs, and minimizes the risk of fraud.

# Scope:

The scope of this project includes the complete lifecycle of an insurance policy, focusing on automating and securing the following key functionalities:

1. **Policy Issuance**:
   * Insurers can issue new policies by specifying the policyholder's details, premium amount (in Ether), coverage amount, and policy duration.
   * Only authorized insurers can issue policies, ensuring secure access control.
2. **Premium Payments**:
   * Policyholders can pay their premiums directly through the smart contract.
   * The contract verifies that the correct premium is paid before updating the policy's status to active.
3. **Claim Submission**:
   * Policyholders can submit claims for events covered by their policies.
   * The smart contract checks whether the claim is valid by ensuring the policy is still active and the claim amount does not exceed the coverage.
4. **Claim Approval and Payout**:
   * Insurers can review and approve claims. Once approved, the smart contract processes the payout by transferring the claim amount (in Ether) to the policyholder.
   * All claim-related transactions are transparent, secure, and traceable on the blockchain.
5. **Blockchain Transparency**:
   * Events such as **PolicyIssued**, **PremiumPaid**, **ClaimSubmitted**, **ClaimApproved**, and **ClaimPaid** are logged to the blockchain for auditability and transparency.
   * Each policyholder, insurer, and auditor can access the entire transaction history, ensuring no data tampering or disputes.
6. **Data Integrity and Security**:
   * Using the Ethereum blockchain ensures that all data regarding policies, payments, and claims are immutable and cannot be altered once recorded.
   * The decentralized nature of the blockchain provides resilience against system failures or data breaches.

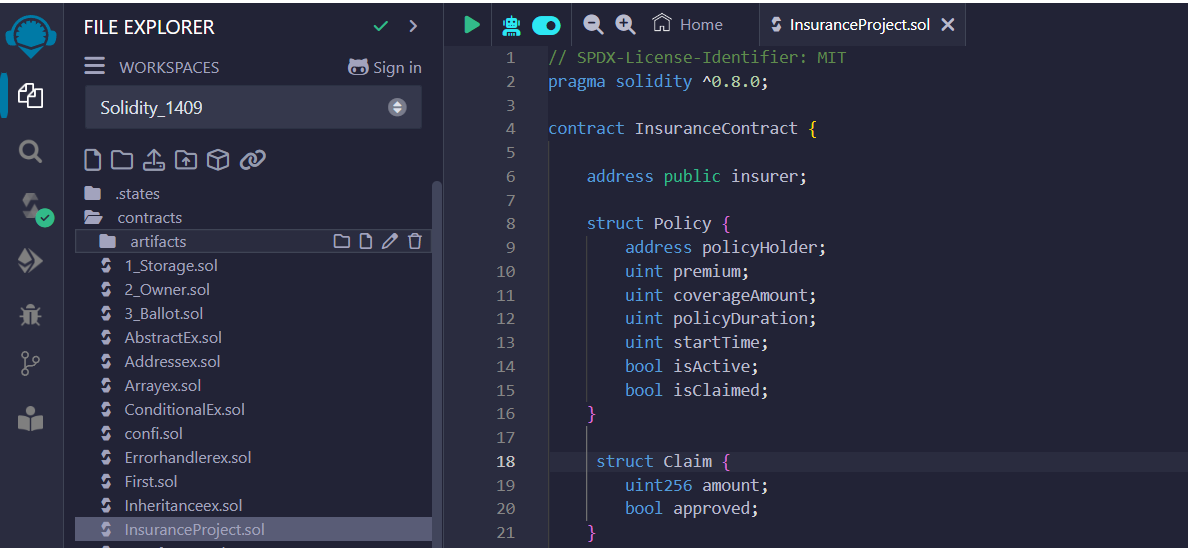
# Execution

## Method description:

|  |  |
| --- | --- |
| Function name | description |
| Policy Issuance (issuePolicy): | Only the insurer can issue policies with defined premium, coverage, and duration. |
| Premium Payment (payPremium) | Policyholders can pay their premium directly to the contract, which updates the policy status. |
| Claim Submission (submitClaim) | Policyholders can submit claims with an amount and reason, only if their policy is active. |
| Claim Approval (approveClaim) | The insurer can review and approve claims. |
| Claim Payout (payClaim): | Once a claim is approved, the insurer can transfer the payout to the policyholder. |

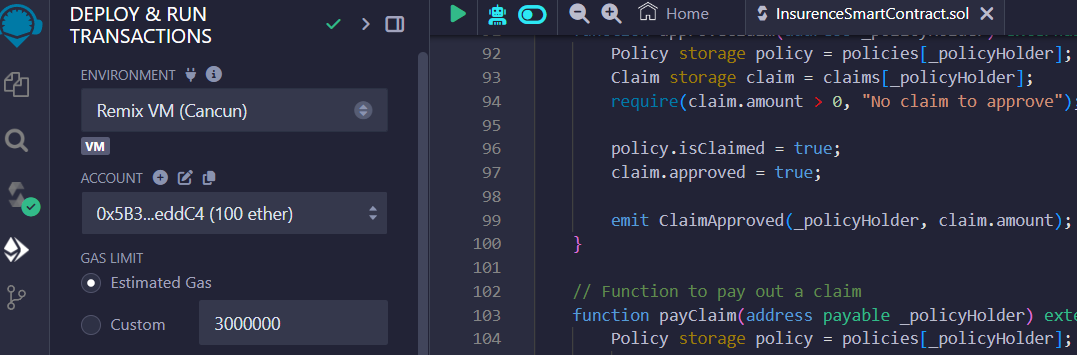
## Compilation

InsurenceSmartContract.sol is deployed

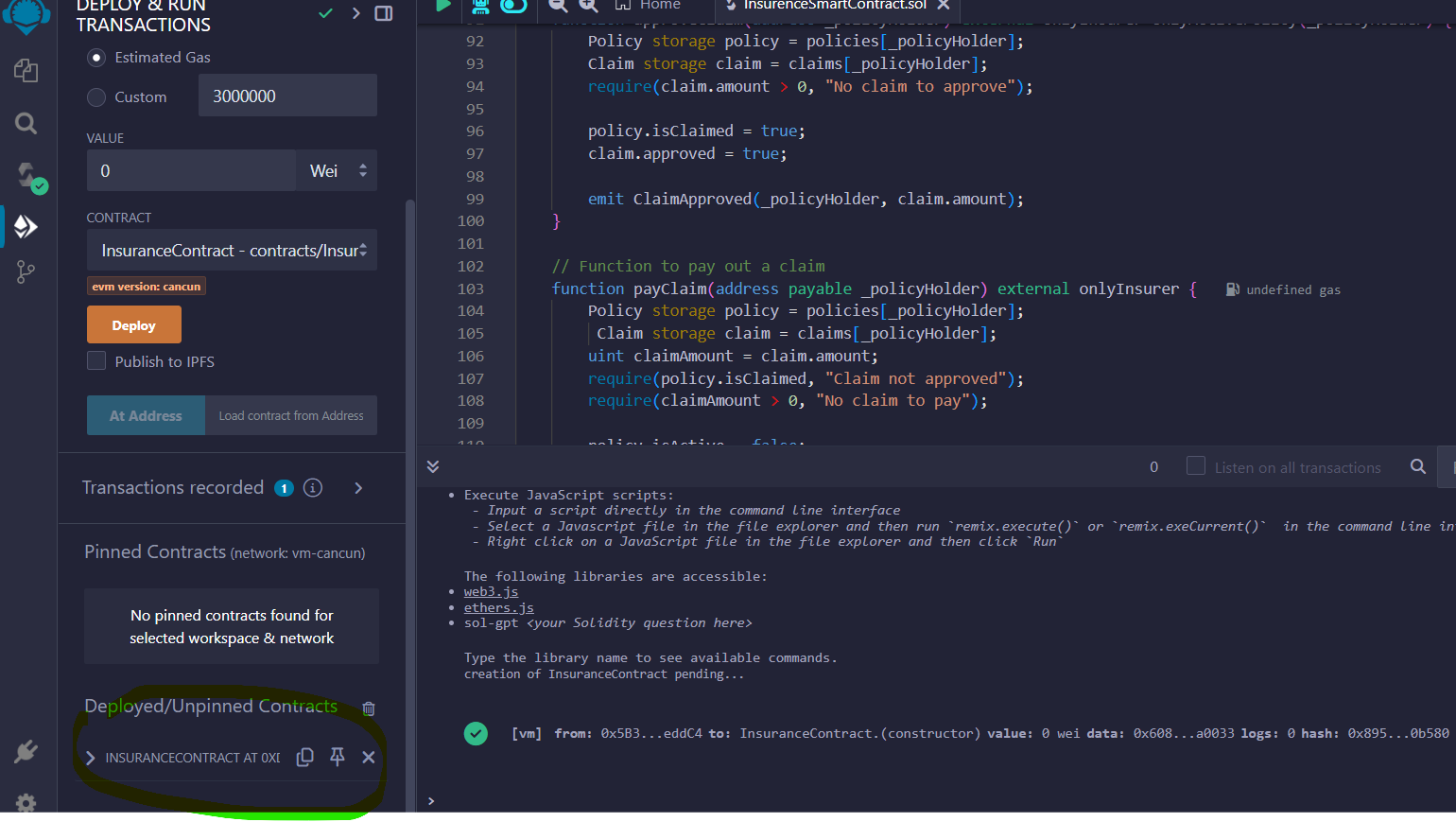


## Deployment:

Deployment will be done by the account id : 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4



Contract is deployed successfully :

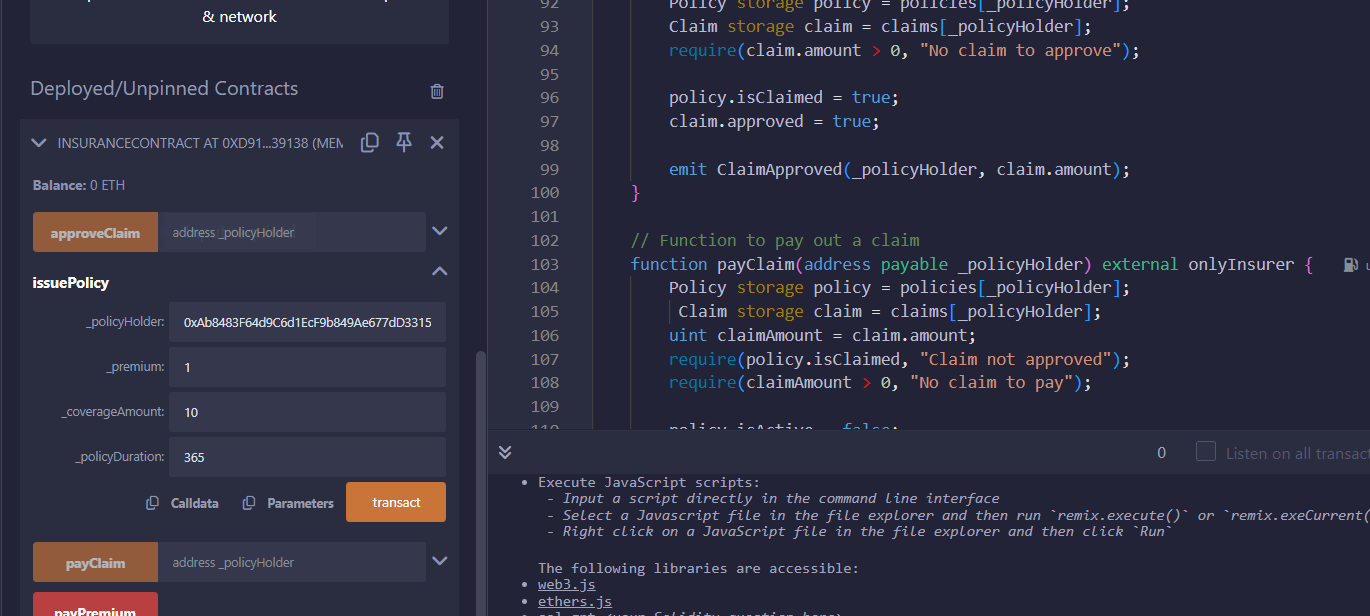


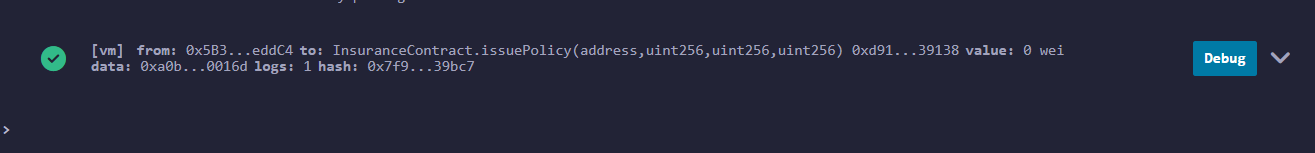
Insurer account id : 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Policy holder account id : 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

Policy process :

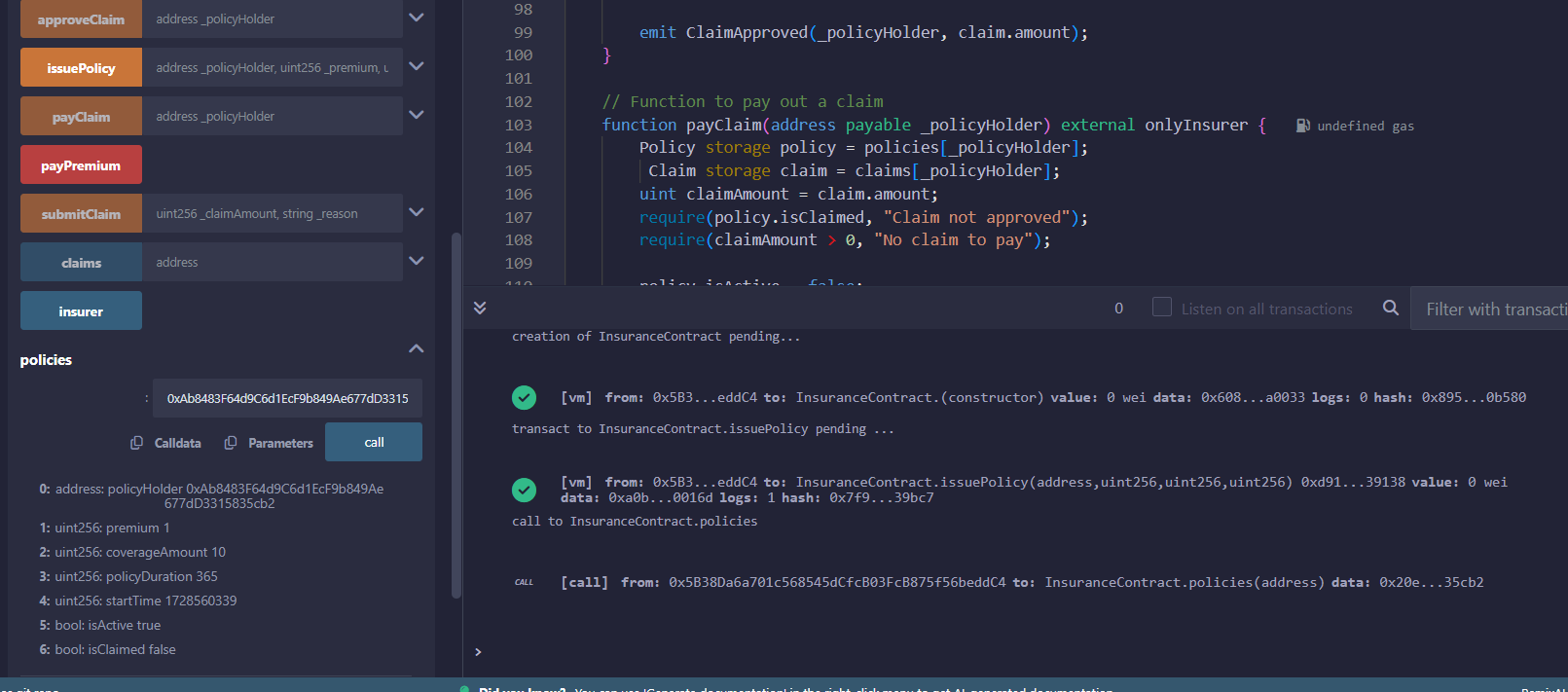
1. Create the policy from insurer account id : 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
2. Click on IssuePolicy button and add the policy holder details:
   1. Policyholder : 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
   2. Premium : 1 (Wei)
   3. converageAmount: 10 (Wei)
   4. policyDuration: 365 days



Click on transact. Policy got added  


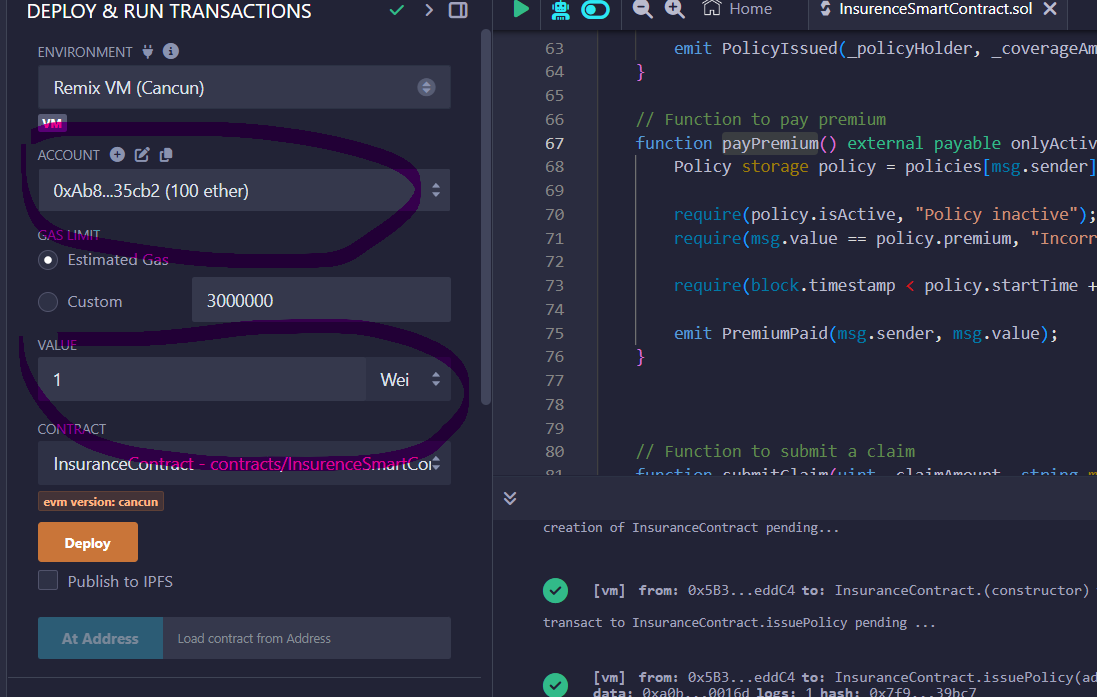
Query the policy holder

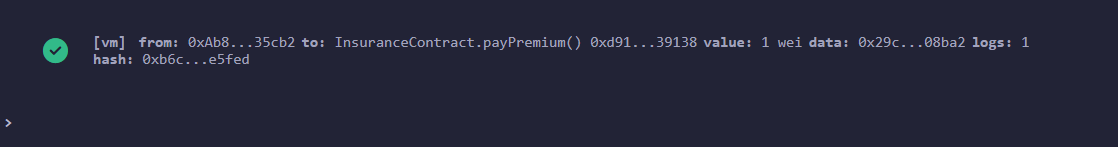
Click on policies button and add the policy holder details 🡪 details shown



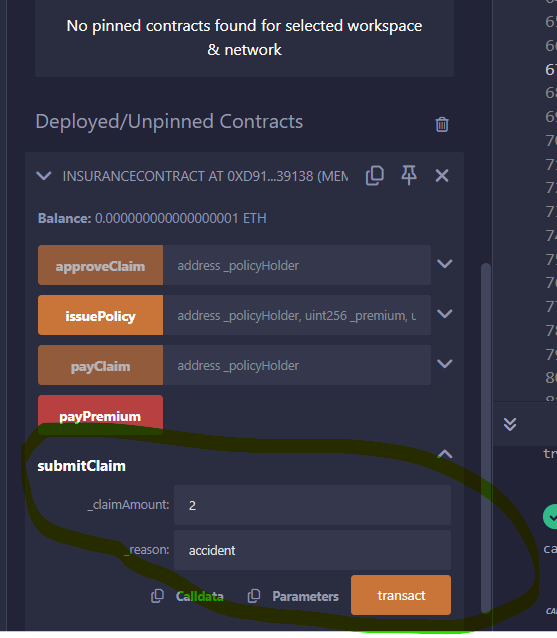
## Add premium and submit claim:

Change account address to policy holder account address change value as 1 Wei and click on **payPremium** button

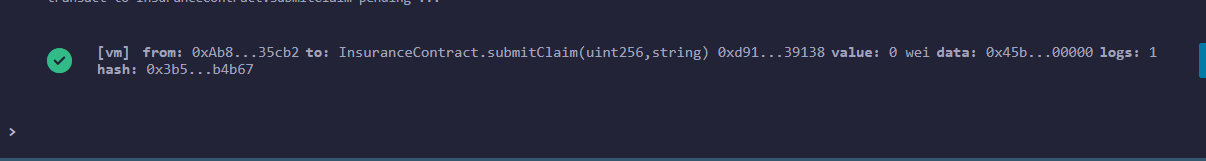




**submitClaim :** Add claimAmount and reason and click on transact



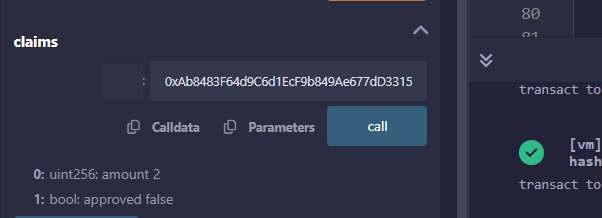
Claim submitted successfully.



## Approve and pay claim amount

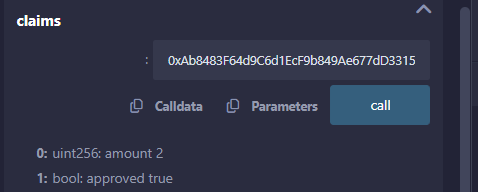
Change account address to insurer address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Claims: Check for claim, currently approve will be false.



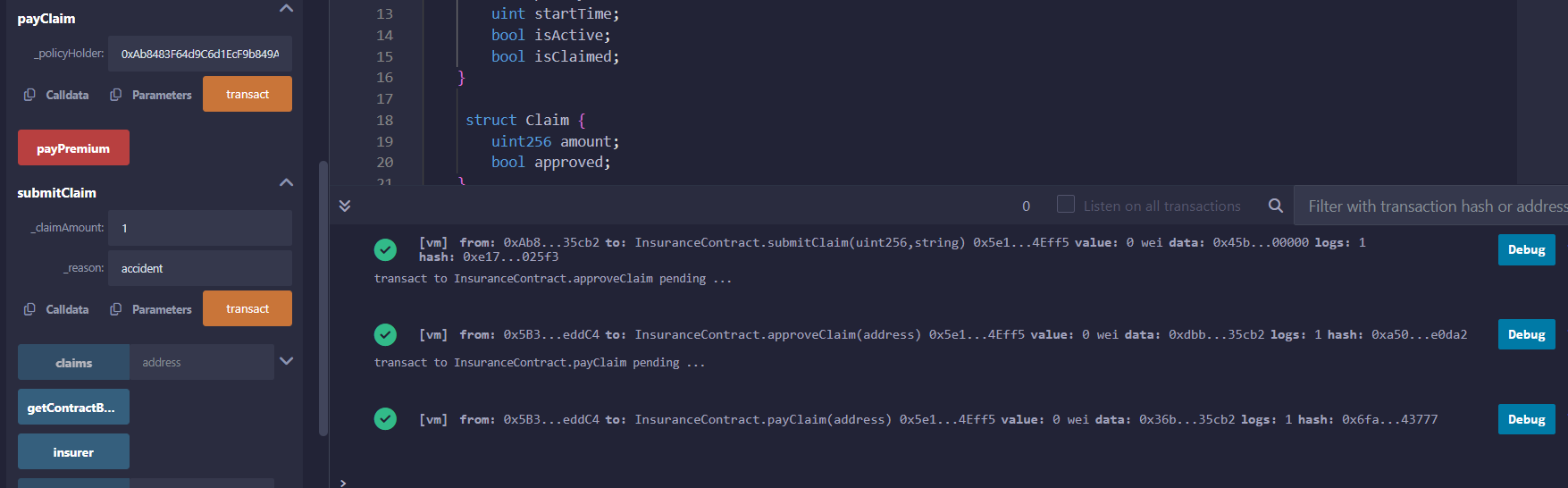
Approve claim: click on **approveClaim** button and add policy holder address : 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 and click on transact button.

Check claim status : click on **claims** button and enter address of policy holder and click on call. Now policy is approved.



Pay claim:

Click on **payClaim** button and enter policy holder address and click on



Claim has been paid to policy holder.