

## Chat Front End API

The Chat's API can be obtained using the EUWA Wrapper Interface, described in [a separate article](#).

### Events List

chatInit

Fired when the chat has been initialized

chatStarts

Fired when the chat starts

chatContinue

Fired when the chat has been continued

chatEnded

Fired when the chat has ended

chatWindowShow

Fired when the chat window has been shown

chatWindowClosed

Fired when the chat window has been closed

chatWindowMinimized

Fired when the chat window has been minimized

contactCenterClose

Fired when the time module has returned not opened exit

### Event Data

```
type eventData = {
  chat: {
    activeUsers: Array<{userId: number, name: string, userType: string}>,
    lastConnectedAgentId: number,
    sessionId: string,
    requestId: number,
    variables: object,
    queueKey: string,
    languageCode: string
  },
  chatter: {
    subject: string,
    name: string,
    email: string,
    hasBeenConnected: boolean,
    hasBeenInConference: boolean
  },
  survey: {
    result: number,
    comment: string
  },
  device: {
    isMobile: boolean
  }
};
```

### Methods

api.startChat()

Similar to clicking the start button. It will open the chat window and show the next view.

`api.minimize()`  
Minimizes the chat window

`api.maximize()`  
Maximizes the chat window

`api.updateVariables(variablesObject)`  
Updates the chat variables

```
api.updateVariables({  
  NewVariable: 'Some Value'  
});
```

`api.updateSystemVariables(variablesObject)`  
Updates system variables (enteredFormName, enteredChatId, enteredFormIssue, selectedQueueKey, timeId2Map)

```
api.updateSystemVariables({  
  enteredFormName: 'Chatter name',  
  enteredChatId: 'chatter@email.com',  
  enteredFormIssue: 'I have issues with my cookies. I ate them all.',  
  selectedQueueKey: 'q_cookies_problems',  
  timeId2Map: 'cookiesQueueWorkingTime'  
});
```

`api.getState()`  
Returns object with minimal chat state.

```
const { isConnected, isEnded, isMinimized } = api.getState();
```