```
use serde::de::DeserializeOwned;
use serde::Serialize;

use crate::client::jsonrpc::ZabbixApiResponse;
use crate::error::ZabbixApiError;
use crate::host::{ZabbixHost, ZabbixHostGroup};
use crate::host::create::{CreateHostGroupRequest, CreateHostRequest};
use crate::item::create::CreateItemRequest;
use crate::item::ZabbixItem;
use crate::trigger::create::CreateTriggerRequest;
use crate::trigger::ZabbixTrigger;
use crate::webscenario::create::CreateWebScenarioRequest;
use crate::webscenario::ZabbixWebScenario;

pub mod jsonrpc;
pub mod v6;
pub mod post;

pub trait ZabbixApiClient {
    /// API: https://www.zabbix.com/documentation/6.0/en/manual/api/reference/apiinfo/version
    fn get_api_info(&self) -> Result<String, ZabbixApiError>;

    fn get_auth_session(&self, login: &str, token: &str) -> Result<String, ZabbixApiError>;

    fn raw_api_call<P: Serialize, R: DeserializeOwned>(&self, session: &str, method: &str, params: &P) -> Res
ult<ZabbixApiResponse<R>, ZabbixApiError>;

    fn get_host_groups<P: Serialize>(&self, session: &str, params: &P) -> Result<Vec<ZabbixHostGroup>, Zabbix
ApiError>;

    /// # get_hosts
    ///
    /// Find zabbix hosts.
    ///
    /// API: https://www.zabbix.com/documentation/6.0/en/manual/api/reference/host/get
    ///
    /// **Example:**
    ///
    /// ```rust
    /// use reqwest::blocking::Client;
    /// use zabbix_api::host::get::GetHostsRequest;
    /// use serde::Serialize;
    /// use zabbix_api::client::v6::ZabbixApiV6Client;
    /// use zabbix_api::client::ZabbixApiClient;
    ///
    /// #[derive(Serialize)]
    /// struct Filter {
    ///   pub host: Vec<String>
    /// }
    ///
    /// let request = GetHostsRequest {
    ///     filter: Filter {
    ///     host: vec!["srv-1203".to_string()],
    ///   },
    /// };
    ///
    /// let http_client = Client::new();
    ///
    /// let zabbix_server = env!("ZABBIX_API_URL");
    ///
    /// let client = ZabbixApiV6Client::new(http_client, &zabbix_server);
    ///
    /// let session = client.get_auth_session("Admin", "zabbix").unwrap();
    /// let hosts = client.get_hosts(&session, &request).unwrap();
    /// ```
    fn get_hosts<P: Serialize>(&self, session: &str, params: &P) -> Result<Vec<ZabbixHost>, ZabbixApiError>;


    /// # get_items
    ///
    /// Find zabbix items.
    ///
    /// API: https://www.zabbix.com/documentation/6.0/en/manual/api/reference/item/get
    fn get_items<P: Serialize>(&self, session: &str, params: &P) -> Result<Vec<ZabbixItem>, ZabbixApiError>;

    fn get_triggers<P: Serialize>(&self, session: &str, params: &P) -> Result<Vec<ZabbixTrigger>, ZabbixApiEr
ror>;
```

```rust
    fn get_webscenarios<P: Serialize>(&self, session: &str, params: &P) -> Result<Vec<ZabbixWebScenario>, ZabbixApiError>;

    fn create_host_group(&self, session: &str, request: &CreateHostGroupRequest) -> Result<u32, ZabbixApiError>;

    /// # create_host
    ///
    /// Create zabbix host.
    ///
    /// API: https://www.zabbix.com/documentation/6.0/en/manual/api/reference/host/create
    ///
    /// **Example:**
    ///
    /// ```rust
    /// use std::collections::HashMap;
    /// use fake::{Fake, Faker};
    /// use reqwest::blocking::Client;
    /// use zabbix_api::host::get::{GetHostGroupsRequest, GetHostsRequest};
    /// use serde::Serialize;
    /// use zabbix_api::client::v6::ZabbixApiV6Client;
    /// use zabbix_api::client::ZabbixApiClient;
    /// use zabbix_api::host::create::{CreateHostGroupRequest, CreateHostRequest};
    /// use zabbix_api::ZABBIX_EXTEND_PROPERTY_VALUE;
    ///
    /// let http_client = Client::new();
    ///
    /// let zabbix_server = env!("ZABBIX_API_URL");
    ///
    /// let client = ZabbixApiV6Client::new(http_client, &zabbix_server);
    ///
    /// let session = client.get_auth_session("Admin", "zabbix").unwrap();
    ///
    /// let filter: HashMap<String,String> = HashMap::new();
    ///
    /// let request = GetHostGroupsRequest {
    ///     output: ZABBIX_EXTEND_PROPERTY_VALUE.to_string(),
    ///     filter
    /// };
    ///
    /// let host_groups = client.get_host_groups(&session, &request).unwrap();
    /// let host_group = host_groups.first().unwrap().clone();
    /// let host_name = Faker.fake::<String>();
    ///
    /// let request = CreateHostRequest {
    ///     host: host_name,
    ///     groups: vec![host_group],
    ///     interfaces: vec![],
    ///     tags: vec![],
    ///     templates: vec![],
    ///     macros: vec![],
    ///     inventory_mode: 0,
    ///     inventory: Default::default(),
    /// };
    ///
    /// client.create_host(&session, &request).unwrap();
    /// ```
    fn create_host(&self, session: &str, request: &CreateHostRequest) -> Result<u32, ZabbixApiError>;

    fn create_item(&self, session: &str, request: &CreateItemRequest) -> Result<u32, ZabbixApiError>;

    fn create_trigger(&self, session: &str, request: &CreateTriggerRequest) -> Result<u32, ZabbixApiError>;

    fn create_webscenario(&self, session: &str, request: &CreateWebScenarioRequest) -> Result<u32, ZabbixApiError>;
}
```