```
use std::fmt::{Display, Formatter};

use serde::de::StdError;
use serde::Deserialize;
use thiserror::Error;

#[derive(Error, Debug)]
pub enum ZabbixApiError {
    #[error("network error")]
    NetworkError(#[from] reqwest::Error),

    #[error("unsupported zabbix api")]
    UnsupportedApiError(#[from] serde_json::Error),

    #[error("zabbix api call error")]
    ApiCallError {
        #[source]
        zabbix: ZabbixError
    },

    #[error("zabbix api bad request error")]
    BadRequestError,

    #[error("zabbix api error")]
    Error
}

#[derive(Deserialize,Debug)]
pub struct ZabbixError {
    pub code: i32,
    pub message: String,
    pub data: String
}

impl Display for ZabbixError {
    fn fmt(&self, f: &mut Formatter<’_>) -> std::fmt::Result {
        write!(f, "[zabbix error] code {}, message ’{}’, data: ’{}’ [/zabbix error]", self.code, self.message
, self.data)
    }
}

impl StdError for ZabbixError {

}
```