use std::fmt::Display;

/\*\*

\* Custom error

\*/

#[derive(Debug)]

pub enum SegDenseError {

IoError(std::io::Error),

Json(serde\_json::Error),

JsonMissingRoot,

JsonMissingObject,

JsonMissingArray,

JsonArraySize,

JsonMissingInputFeature,

}

impl Display for SegDenseError {

fn fmt(&self, f: &mut std::fmt::Formatter<'\_>) -> std::fmt::Result {

match self {

SegDenseError::IoError(io\_error) => write!(f, "{}", io\_error),

SegDenseError::Json(serde\_json) => write!(f, "{}", serde\_json),

SegDenseError::JsonMissingRoot => {

write!(f, "{}", "SegDense JSON: Root Node note found!")

}

SegDenseError::JsonMissingObject => {

write!(f, "{}", "SegDense JSON: Object note found!")

}

SegDenseError::JsonMissingArray => {

write!(f, "{}", "SegDense JSON: Array Node note found!")

}

SegDenseError::JsonArraySize => {

write!(f, "{}", "SegDense JSON: Array size not as expected!")

}

SegDenseError::JsonMissingInputFeature => {

write!(f, "{}", "SegDense JSON: Missing input feature!")

}

}

}

}

impl std::error::Error for SegDenseError {}

impl From<std::io::Error> for SegDenseError {

fn from(err: std::io::Error) -> Self {

SegDenseError::IoError(err)

}

}

impl From<serde\_json::Error> for SegDenseError {

fn from(err: serde\_json::Error) -> Self {

SegDenseError::Json(err)

}

}