

- 1) How does the deserialization on the RPC endpoint work (e.g Github links to code regarding deserialization, architecting charts of the back end, etc)?
 - a) Are there additional steps in the BCS deserialization? Walk us through BCS Writer <https://github.com/MystenLabs/sui/blob/main/sdk/bcs/src/index.ts#L634>
 - b) How different is this BCS from the standard Libra implementation
Walk me through TransactionBlock Serialization:
<https://github.com/MystenLabs/sui/blob/4e7773017e1c05e191fac09e34529b656bf99311/sdk/typescript/src/builder/TransactionBlockData.ts#L227C2-L227C2>
- 2) Explain how PrepareTransactions() within TransactionBlock works (e.g the purpose of the function if not self-explanatory, paradigms it uses).
 - a) <https://github.com/MystenLabs/sui/blob/4e7773017e1c05e191fac09e34529b656bf99311/sdk/typescript/src/builder/TransactionBlock.ts#L499>
 - b) We'd like more clarity on what it's supposed to do
 - i) It scans through the transactions and filters out certain transactions that meet a criteria
 - (1) E.g., it filters out move module transactions and resolves those.
 - ii) What does "[resolving](#)" mean in this context?
- 3) Help with figuring out how txByte data is formed
 - a) <https://github.com/MystenLabs/sui/blob/4e7773017e1c05e191fac09e34529b656bf99311/sdk/typescript/src/builder/TransactionBlockData.ts#L164>
- 4) Any endpoints within the Sui RPC documentation that I could look into more to help with the current issue (e.g help with figuring out how txByte data is formed)?
- 5) Any other suggestions / resources for looking into a solution for the current issue?
- 6) Give some high end explanations for the Typescript SDK (e.g paradigms used, why is it architected the way it is, etc).
- 7) If you were to reconstruct the Typescript SDK, how would you go about doing so (e.g different paradigms)?