```python
# Packet Structure (request.py)
# Bach Vu
# 01/08/2020

from packet import *

class DT_Request(DT_Packet):
    ErrorMessage = [
        "Expecting received packet have MagicNum 0x497E.",
        "Expecting received packet have packetType 0x0001.",
        "Undefined outupt Type [date/time]?",
        "Packet header is shorter than expected"
    ]
    def __init__(self, mode, head_info=None):
        self.requestType = mode # 0x0001 or 0x0002
        if head_info is None:
            super().__init__(0x0001)
        else:
            self.MagicNum   = head_info[0]
            self.packetType = head_info[1]

    def __repr__(self):
        out = "<Magic: {}> <packetType: {}> <requestType: {}>\n"
        out = out.format(hex(self.MagicNum), DT_Packet.DT_hex(self.packetType), DT_Packet.DT_hex(self.requestType))
        out += type(self).__name__ + " with request type: " + ("DATE" if self.requestType==1 else "TIME")
        return out

    def header_errorCode(self):
        error_code = 0
        if self.MagicNum != 0x497E:
            error_code = 1
        elif self.packetType != 0x0001:
            error_code = 2
        elif self.requestType < 0x0001 or self.requestType > 0x0002:
            error_code = 3

        return error_code

    def encodePacket(self):
        # Error check
        check = self.isValid()
        if check != 0:
            return check

        # Header
        header = ""
        header += DT_Packet.intToBinStr(self.MagicNum,16)
        header += DT_Packet.intToBinStr(self.packetType,16)
        header += DT_Packet.intToBinStr(self.requestType,16)
        header  = int(header, 2).to_bytes(6, byteorder='big')

        # Pack
        packet = bytearray()
        packet += header
        return packet

    @staticmethod
    def decodePacket(packet):
        if len(packet) < 6:
            return 4

        magic    = DT_Packet.byteArrToInt(packet[0:2])
        packType = DT_Packet.byteArrToInt(packet[2:4])
        mode = DT_Packet.byteArrToInt(packet[4:6])
        requestPack = DT_Request(mode)

        # Error check
        param = [magic, packType]
        requestPack = DT_Request(mode, tuple(param))
        check = requestPack.isValid()
        if check != 0:
            return check
        return requestPack

```